Detecting IP prefix hijack events using BGP activity and AS connectivity analysis

BY

Hussain Hameed Alshamrani

A thesis submitted to the University of Plymouth in partial fulfilment for the degree of

Doctor of Philosophy

Centre for Security, Communications and Network (CSCAN)

Plymouth University

February 2017

Copyright

Abstract

The Border Gateway Protocol (BGP), the main component of core Internet connectivity, suffers vulnerability issues related to the impersonation of the ownership of IP prefixes for Autonomous Systems (ASes). In this context, a number of studies have focused on securing the BGP through several techniques, such as monitoring-based, historical-based and statistical-based behavioural models. In spite of the significant research undertaken, the proposed solutions cannot detect the IP prefix hijack accurately or even differentiate it from other types of attacks that could threaten the performance of the BGP. This research proposes three novel detection methods aimed at tracking the behaviour of BGP edge routers and detecting IP prefix hijacks based on statistical analysis of variance, the attack signature approach and a classification-based technique.

The first detection method uses statistical analysis of variance to identify hijacking behaviour through the normal operation of routing information being exchanged among routers and their behaviour during the occurrence of IP prefix hijacking. However, this method failed to find any indication of IP prefix hijacking because of the difficulty of having raw BGP data hijacking-free.

The research also proposes another detection method that parses BGP advertisements (announcements) and checks whether IP prefixes are announced or advertised by more than one AS. If so, events are selected for further validation using Regional Internet Registry (RIR) databases to determine whether the ASes announcing the prefixes are owned by the same organisation or different organisations. Advertisements for the same IP prefix made by ASes owned by different organisations are subsequently identified as hijacking events. The

proposed algorithm of the detection method was validated using the 2008 YouTube Pakistan hijack event; the analysis demonstrates that the algorithm qualitatively increases the accuracy of detecting IP prefix hijacks. The algorithm is very accurate as long as the RIRs (Regional Internet Registries) are updated concurrently with hijacking detection. The detection method and can be integrated and work with BGP routers separately.

Another detection method is proposed to detect IP prefix hijacking using a combination of signature-based (parsing-based) and classification-based techniques. The parsing technique is used as a pre-processing phase before the classification-based method. Some features are extracted based on the connectivity behaviour of the suspicious ASes given by the parsing technique. In other words, this detection method tracks the behaviour of the suspicious ASes and follows up with an analysis of their interaction with directly and indirectly connected neighbours based on a set of features extracted from the ASPATH information about the suspicious ASes. Before sending the extracted feature values to the best five classifiers that can work with the specifications of an implemented classification dataset, the detection method computes the similarity between benign and malicious behaviours to determine to what extent the classifiers can distinguish suspicious behaviour from benign behaviour and then detect the hijacking. Evaluation tests of the proposed algorithm demonstrated that the detection method was able to detect the hijacks with 96% accuracy and can be integrated and work with BGP routers separately.

Acknowledgement

Regarding the occasion of finishing the research, I would like to thank my mother and father for their patience while I was far away for obtaining my PhD certificate. I would also like to thank my supervisors for their useful guidance, and my wife and children for being here to support me during the research period. Special thanks to Bogdan Ghita for his guidance and corrections to my papers and thesis. I also would like to acknowledge Rana Moyeed and Irene Kaimi for their helpful statistical guidance, and my friend Ali Hussain Al-Timemy for his advice during using machine learning as a tool to detect IP prefix hijacking. Finally, many thanks to all my colleagues for their technical assistance and moral support.

Table of Contents

Glossary

ABEs	Abnormal BGP Events
APNIC	Asia-Pacific Network Information Centre
ARIN	American Registry for Internet Numbers
AS	Autonomous System
ASN	Autonomous System Number
ASNs	Autonomous System Numbers
BGP	Border Gateway Protocol
BSA	Binary Search Algorithm
CA	Certificate Authorisation
CART	SimpleCart
CIDR	Classless Internet Domain Routing
CRL	Certificate Revocation List
CSV	Comma-Separated Values
DDoS	Distributed Denial of Service
EBGP	External Border Gateway Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
FE	Feature Extractor
HMM	Hidden Markov Model
IANA	Internet Assigned Numbers Authority
IBGP	Internal Border Gateway Protocol
ICMP	Internet Control Message Protocol
IRR	Internet Routing Registry
IRV	Internet Route Verification

ISP	Internet Service Provider
k-NN	k-Nearest Neighbour
ML	Machine Learning
MOAS	Multiple-Origin Autonomous System
MOR	Multi-class Odds Ratio
mRMR	minimum Redundancy Maximum Relevance
MRT	Multi-threaded Routing Toolkit
NAV	Network Analysis and Visualization
NB	Naive Bayes
Nemecis	NEtwork ManagEment and Configuration System
NLF	Non-Linear Filtering
NLRI	Network Layer Reachability Information
OSPF	Open Shortest Path First
PAL	Prefix Assertion List
PCA	Principal Component Analysis
PGBGP	Pretty Good BGP
PHAS	Prefix Hijack Alert System
PKI	Public Key Infrastructure
psBGP	pretty secure Border Gateway Protocol
RBF	Radial Basis Function
RF	Random Forest
RIB	Routing Information Base
RIDB	Routing Information Database
RIPE-RIS	Reseaux Internet Protocol European's-Routing Information

	Service
RIPE's RPSS	Reseaux Internet Protocol European's Routing Policy System Security
RIR	Regional Internet Registry
RIS	Routing Information Service
ROAs	Route Origin Authorisations
RP	Replying Party
RPKI	Resource Public Key Infrastructure
RRCs	Remote Route Collectors
RRI	Registered Routing Information
SASL	Suspicious Autonomous System List
sBGP	secure Border Gateway Protocol
soBGP	secure origin Border Gateway Protocol
SVM	Support Vector Machine
TERRAIN	Testing and Evaluation of Routing Robustness in Assurable Inter-domain Networking
TTL	Time To Live

Author's Declaration

At no time during the registration for the research degree of Doctor of Philosophy has the author been registered for any other University award. In addition, work submitted for this research degree at Plymouth University has not formed part of any other degree either at Plymouth University or at another establishment. Moreover, this study was financed with funding from Jazan University and Saudi Cultural Bureau. Relevant scientific conferences were regularly attended at which the work presented. Contacts from Saudi Cultural Bureau provided technical advice and guidance, especially in the early stages of the project.

Sign…….

Date…….

List of Figures

List of Tables

Chapter : 1 Introduction

BGP version 4 is currently the standard protocol for core Internet interconnection. The BGP was initially designed without security in order to make the Internet faster and more effective. Together with its popularity, the BGP also has a number of drawbacks linked to its early adoption and age, particularly drawbacks linked to peer-trust and policy-based routing. However, its flexibility allows attackers to exploit its flaws by compromising the BGP and path selection decision policies [1]. If the vulnerabilities are classified in the BGP, they are found in the area of open message and update message. However, update message is special because it is considered the heart of routing table changes in the BGP. The first part of the BGP update message issue is related to misconfiguration, while the second refers to manipulating a sequence of operations applied on the BGP, such as selection routing processes and BGP route export policy, which is considered a flexible feature for engineering purposes.

There are some inherent design principles that make BGP vulnerable: lack of integrity validation, freshness and origin authentication of messages, and no validation of AS authority or path authenticity [2]. Nevertheless, the BGP remains the protocol of choice for core Internet interconnectivity because it is considered the strongest and most flexible routing protocol in terms of functionality. However, its flexibility also makes it vulnerable to attack from untrusted routers.

Although a number of BGP security issues have been identified for almost two decades, the protocol is still vulnerable to IP prefix attacks [3]. These issues facilitate serious attacks and open the door to other types of attack, such as spam attacks [4], traffic interception and DDoS

[5]. Direct financial benefit has also been a motive for attacks: In February 2014, Valadon and Vivet claimed that an attacker redirected some cryptocurrency miners to their own mining pool [6]. Moreover, route leaking occurs when a peer incorrectly advertises a varying degree of prefixes from its RIB (Routing Information Base) [7]. Power outages and cable cuts are not attacks, but they do affect routing stability and can also impact network infrastructure. For example, the Moscow blackout damaged the whole European and Asia-Pacific region. Conversely, cable cuts can cause more specific problems, such as the disabling of the critical egress links of a given region [7]. The BGP is also susceptible to worms, which affect the stability of routers. The Code Red worm, which affects the stability of routers as in [8], is considered the most effective type of worm attack. For prefix hijacking, attacks happen when a speaker infiltrates a BGP prefix announcement by impersonating the original ownership of others' ASes (Autonomous Systems). Finally, a hijacker can impersonate ownership of an IP prefix, which is the main area of this research.

Different prevention-based solutions anticipated to secure the BGP and detect IP prefix hijacking. However, Vervier et al. indicate that prevention solutions that use ASes and BGP speaker authentication and verification are still facing large-scale deployment issues [5]. Due to several reasons, issues on large routing systems or impractical approaches like S-BGP [9], threats continue to exist [10]. Moreover, Wubbeling et al. pointed out that security based on origin authentication and asymmetric encryption is not feasible because the required underlying mechanisms are not yet implemented broadly [10]. In addition, the RPKI (Resource Public Key Infrastructure) system is one of the IP prefix hijacking security systems put into place to prevent BGP route hijacking. The system is based on tracing the hierarchical relationships of the address space, which are given by the IANA (Internet Assigned Numbers Authority), RIRs and big ISPs to customers, to AS origins. The system checks the origin

ASes or authorised origin ASes to announce specific IP prefixes. The system uses ROAs (Route Origin Authorisations), which contain IP prefixes and their authorised ASes. ROAs provide the means for verifying if an IP prefix holder has authorised an AS to originate one or more IP prefixes. ROAs are cryptographically signed and published in repositories [11]. Routers can download these repositories using a trusted tool and then upload them into routers [12].

A traditional solution such as using rule-based, prefix-owner-centric, origin-set monitoring, origin ASes and authentication-based IP prefixes, was employed by prior research to detect IP prefix hijacking based on originality authentication [13], [14], [15] and to monitor the stability of the encompassing routers. However, Vervier et al. noted that solutions based on monitoring anomalies to detect IP prefix hijacking still suffer from high false positive rates [5]. Other solutions analyse routing tables (table-based) in order to detect IP prefix hijacking, but some organisations refuse to provide their routing tables; therefore, these solutions encounter some difficulties in evaluating their experiments [16]. In addition, some solutions based on anomaly detection have been used to detect different anomalies, including IP prefix hijacking. However, these solutions cannot reliably distinguish IP prefix hijacks from normal events, such as power cut offs and submarine cable cuts [7].

1.1 Aim and objectives

The aim of the this thesis is to design, investigate, and benchmark a set of novel approaches for detecting IP prefix hijacking attacks based on statistical analysis of routing communication between routers, combined with attack signature and connectivity-related

metrics for suspicious ASes. To achieve this aim, the following research objectives were identified:

a) To identify the state of the art in the architecture, policy, communication, vulnerabilities, and built-in security of BGP.

b) To investigate the concept of IP prefix hijacking, particularly focusing on past incidents and their impact on the network, their footprint on the exchanged BGP update messages, and the connectivity between participating or affected routers.

c) To review the current state-of-the-art research in the area of IP prefix hijacking prevention and detection.

d) To investigate how BGP update messages and statistical analysis of router behaviour can be used as inputs for detecting IP prefix hijacking incidents.

e) To improve and expand on the method proposed by cross-validating the results with the content of RIR databases.

f) To investigate how the accuracy of IP prefix hijacking detection can be improved by using the outputs of the methods proposed under objectives 4 and 5 and classifying corresponding BGP update packets as benign or malicious.

g) To evaluate the efficiency of the developed detection methods based on data from historical IP prefix hijacking incidents.

h) To conceptually integrate the proposed methods into a BGP routing architecture based on collaborative work to alleviate the impact of a IP prefix hijacking incident onto the wider Internet.

1.2 Thesis contents

This thesis discusses flaws related to the BGP update message, specifically focusing on detecting IP prefix hijacking, and then proposes the appropriate methods to detect IP prefix hijacking while achieving the objectives discussed in section 1.1. Chapter 1 reflects the content and weakness of the BGP and then points out the two methods that have been proposed to detect the IP prefixes. The chapter also discusses the aim and objectives that has to be achieved by the end of the research. In addition, Chapter 1 presents the results of the methods. Chapter 2 describes the background of the BGP, its contents, how it works, and its vulnerabilities; it then talks about already built-in security mechanisms. Furthermore, the chapter briefly discusses IP prefix hijacking that has already affected the BGP and then explains the process of IP prefix hijacking, giving an example to clarify the picture. Chapter 2 also discusses previous solutions that have tried to secure the BGP based on prevention and detection techniques and evaluates their methodology and results. The aim of the chapter is to review the previous literature and propose a proper, novel method based on their limitations to secure the BGP. The chapter divides security solutions into two types: prevention and detection. The detection solutions aim to detect IP prefix hijacking after it occurs, while the prevention solutions try to prevent IP prefix hijacking before it takes place. For example, the RPKI (Resource Public Key Infrastructure) and the combination of rule-based and packet-based techniques are categorised in the first type of solutions, while sBGP, soBGP and

psBGP are categorised in the second type of solutions. These solutions will be discussed in detail in the chapter.

In consideration of the advantages and limitations of previous solutions, Chapter 3 discusses a detection method using statistical analysis of variance to detect IP prefix hijacking. In other words, the chapter performs data processing and an analysis of BGP updates to determine the most appropriate method that can be used to avoid the mistakes, which occurred in the previous solutions. This chapter illustrates the data sources that were used and the reasons for selecting specific sources. Moreover, raw data from BGP updates need some preparation and organisation to suit the methodology of the detection method. The chapter surveys and studies the data that can be used to secure the BGP, especially for detecting IP prefix hijacking and applying the data in two-dimensional graphs.

Chapter 4 proposes a novel self-checking, signature-based and RIR verification-based method to detect IP prefix hijacking. As a case study, BGP update messages will be collected from February 24, 2008, when Pakistan Telecom intended to restrict local access to YouTube, but the advertised BGP update messages blocked access to YouTube [17] for approximately two hours [18]. BGP update messages are downloaded from different routers in the Route Views Archive Project [21]. The detection method traces origin ASes and their actual IP prefixes per 15-minute time slots. It is composed of two parts: the first part receives downloaded updates, searches for the signature of hijacking, and then passes the results as suspicious events on to the RIRs verification-based part to judge the BGP announced packets. The self-checking signature-based method is based on data reduction technique and a binary search algorithm and aims to improve the processing speed for detecting IP prefix hijacking

events. The results of the detection method are explained and evaluated at the end of the chapter.

Chapter 5 discusses the second method used in this thesis, which is composed of two main components: self-checking signature-based and connectivity-based. The self-checking signature-based component traces the suspicious ASes existing in BGP updates and sends them as inputs to the connectivity-based component. The method uses a classification technique to detect IP prefix hijacking. The connectivity-based component extracts features based on the connectivity behaviour of suspicious ASes that it receives from the signature-based component. In other words, the method will have a dataset based on the behaviour of suspicious AS connectivity. The similarity behaviour of benign ASes and malicious ASes is calculated independently, and the difference in behaviour between the two classes of ASes is also calculated in order to probe the quality of the data and predict the classifiers' results. The behavioural dataset is given to five leaning classifiers (J48, k-Nearest Neighbours, Naïve Bayes, Classification And Regression Tree and Random Forest) based on its characteristics. Each classifier creates a classification model based on the extracted features and the values of the classifiers' parameters. As a result, unseen cases of hijacking will be detected based on the model created by classifiers during classifications.

In Chapter 6, a proposed architecture for how the three detection methods have to be linked to the BGP to activate collaboration between routers instead of relying on centralised functional structures is discussed. The chapter starts by presenting the architecture method and then discusses the functionality of the proposed detection methods after they are linked to the BGP collaboratively. Finally, the chapter also describes the responsibility of the network operators when they receive an alarm with a hijack. In other words, removing the hijacking is

the responsibility of the network operators, not the detection method. However, this thesis suggests the most appropriate way to prevent the bogus routes from spreading out.

The last chapter summarises how the three proposed detection methods work and discusses their advantages and disadvantages. The chapter also discusses difficulties that the detection methods face. Finally, the chapter compares the detection methods with previous BGP security solutions.

Chapter : 2 BGP security and IP prefix hijack attacks

This chapter discusses the BGP in terms of its architecture and security. First, the chapter provides an overview of the BGP, including its architecture and communication of BGP messages, weaknesses that threaten its security, and the already existing BGP security mechanisms designed to protect router privacy, such as filtering BGP updates coming from unknown neighbours (routers). After presenting a general background on BGP architecture and security, the chapter will focus on the more significant vulnerabilities that threaten the security of the BGP – namely, IP prefix hijacking. This type of attack will be studied from two angles: its history and occurrence. The chapter will talk about the proposed solutions for securing or detecting IP prefix hijacking, such as sBGP, spBGP, the rule-based method and the historical-based method. The solutions will be discussed from different aspects: mechanism, experiment (if the solution is practical not theoretical), result and finally evaluation. At the end, the proposed approaches the solutions are based on will be compared in terms of their strengths and weaknesses.

The chapter is organised as follows: Section 2.1 presents the BGP's background, including BGP architecture and communication, vulnerabilities and security; section 2.2 gives a brief history and discusses the process of IP prefix hijacking; section 2.3 is allocated to previously proposed security solutions for securing the BGP, while section 2.4 talks about the previous detection solutions used to detect IP prefix hijacking in the BGP; section 2.5 compares the two security approaches, prevention-based and detection-based, to select the most appropriate one used in the thesis to detect IP prefix hijacking; lastly, section 2.6 provides a summary of the chapter.

2.1 BGP background

This section discusses the BGP from three perspectives: components and functionality, vulnerabilities and built-in security mechanisms. Section 2.1.1 focuses in particular on BGP messages and their functions to establish connections and update the routing tables among routers. The section will also discuss some parts that routers consist of and how the information could be used to support IP hijack detection. Section 2.1.2 talks about the events that can affect the BGP, while section 2.1.3 concentrates on built-in security methods, such as using MD5 and filtering out received packets.

2.1.1 BGP architecture and communication

The BGP4 (Border Gateway Protocol) is an inter-domain protocol that connects different ASes so they can work together as one big network. In other words, it is used to connect routers over the Internet. This protocol consists of four messages: open, keepalive, notification, and update. The open and keepalive messages are used to establish BGP sessions using the TCP and to monitor the live connection between two connected routers. The notification messages are used for the notification of errors, while the update messages are used for populating and updating routing tables. Each message consists of different fields plus header fields. The header fields have Marker, Length and Type of specific messages. Marker is used to detect loss of synchronisation between a pair of BGP peers and to authenticate incoming BGP messages, while Length and Type to carry the size and the type of the message The open message has six unique, independent fields: (1) Version, which shows the version of the currently used BGP; (2) My Autonomous System Number, which indicates the Autonomous System Number (ASN) of the sender; (3) Hold Time, which

carries the value of connection repose in seconds – if a receiver does not accept the connection in 3 seconds as a maximum period, it will be rejected; (4) BGP Identifier, which is responsible for storing the identification of the sender; (5) Optional Parameter Length, which shows the total length of optional parameters carried in the announcement by a sender; and (6) Optional Parameter, which dictates the optional parameters themselves. The BGP uses the TCP while establishing BGP sessions to send and receive BGP packets to and from neighbours. As a result, the BGP inherits all weaknesses available in the TCP. However, the TCP and other three messages fields are out of the scope and the most important fields to this research are the ones belonging to the update message, because data in these fields are exchanged periodically between routers (direct and indirect neighbours) and are directly related to IP prefix hijacking. These fields include Unfeasible Route Length, Withdrawn Routes, Total Path Attribute Length, Path Attributes and Network Layer Reachability Information. Unfeasible Routes Length indicates the total length of withdrawn routes. Withdrawn Routes contains a list of IP address prefixes for the routes being withdrawn from service. Total Path Attribute Length includes the total length of attributes sent in the update message, while Path Attributes contain the attributes themselves. Network Layer Reachability Information contains a list of IP address prefixes that can be reached.

The BGP is classified as a path vector protocol, but it can work with a distance vector and link state routing protocols. Routers that have the BGP protocol can speak to each other and update their routing tables via update messages within or outside of ASes using either a EBGP (External Border Gateway Protocol) or IBGP (Internal Border Gateway Protocol) path. The RIB (Routing Information Base) is the repository where all routing protocols place their received routes. The RIB has three databases: Adj-RIBs-In, Adj-RIBs-Loc and Adj-RIBs-Out. Routes are stored in Adj-RIBs-In whenever a routing protocol learns a new

route. When a destination becomes unreachable, the route is marked unusable and later removed from the RIB. Adj-RIBs-Loc contains local routing information. Speakers (routers) apply local BGP policies to routing information already located in Adj-RIBs-In and store it in Adj-RIBs-Loc as the valid and best routes. The routing information is taken and stored in an Adj-RIBs-Out storage until ready for advertisement. This process is based on what is known as the BGP Decision Process [22].

2.1.2 Vulnerabilities

The BGP is vulnerable to traffic interception and DDoS (Distributed Denial of Service) attack because it uses the TCP while establishing neighbouring sessions with speakers (edge routers) [4], [5]. Routers in different ASes but connected to each other are called edge routers. Routers with BGP use the TCP when they want to establish BGP sessions and connect to other routers. As a result, the BGP is weak because its infrastructure for establishing BGP sessions is based on the TCP. In other words, any weaknesses related to the TCP will be automatically inherited by the BGP. For example, DDoS attacks can happen to both the TCP and the BGP. However, the BGP does not need to flood the bandwidth or resources of a targeted system directly; but when a hijacker impersonates other IP prefixes, routers automatically search for the best path and try to reach the destination, which impacts the routing table routes at the same time. The BGP can also be affected by worm attacks, such as Nimda, Slammer and Code Red [7]. Worms can only affect the stability of the routers and make the network very slow. All weakness issues existing in the TCP can be inherited in the BGP, but these issues and worm attacks are outside of the scope of this thesis.

IP prefix hijacking is considered the most serious issue to threaten the BGP because it encroaches on the rights and privacy of others. IP prefix hijacking attacks occur when a speaker (hijacker) injects a BGP prefix announcement that impersonates the original owner of other ASes (victims). Prefix hijacking affects the stability of the routers, as worm attacks do, and controls traffic flows, which lead to false traffic redirection. The hijacker performing an IP prefix hijacking attack can also withdraw active routes and make a specific network unreachable without the victim's knowledge. IP prefix hijacking can also result in the blocking of services on other servers such as the incident of YouTube and Pakistan that occurred in 2008. Attackers can use this type of attack for illegitimate operations when they want to hide their identities. The most significant vulnerability of the BGP is that when a victim router receives a fake route, it cannot detect the impersonation. The attacker can also withdraw the routes of other organisations, which leads to blocking services that are provided by service providers.

Moreover, route leaking can happen when a peer incorrectly advertises a varying degree of prefixes from its RIB [7]. Attackers can use redirected BGP traffic in order to steal money. In February 2014, Valadon and Vivet admitted that an attacker had redirected cryptocurrency miners to their own mining pool [6].

2.1.3 Built-in security

The BGP includes its own methods for controlling communication between and privacy among routers; such methods, such as router hardening, generalised TTL, route dampening, maximum prefix limiting, limiting AS_PATH length and prefix filtering [23], and MD5, are gradually modified and added to the BGP. Cisco uses inbound route filtering within its route

policy filtering [24] to secure the BGP and control the receiving of updates from untrusted routers. Filtering routes allows the network operator to determine which neighbours to trust and accept advertisements from. However, this technique is not feasible because BGP routers do not have knowledge of how their direct neighbours filter their indirect neighbours or to what extent indirect neighbours trust each other. In other words, indirect neighbours could affect the routers not linked to them directly because routers work collaboratively with both direct and indirect neighbours but filtering only controls direct connections among routers. As a result, implicit trust among connected ASes might result in the indirect injection of invalid routes, which can in turn reach other ASes [10]. This filtering affects the communication flexibility of the BGP and only controls incoming and outgoing data between neighbours. The BGP also inherits security techniques used in the TCP, such as MD5. However, the problem of using MD5 is that it only provides peer authentication (BGP open message), not BGP update message authentication.

2.2 IP prefix hijacking

IP prefix hijacking is considered a serious issue that threatens the security and stability of the BGP when a neighbour impersonates ownership of other routers' IP prefixes. Any router on the Internet can announce fake routes and claim ownership of them. This approach is called IP prefix hijacking of other routers' prefixes. This section has two subsections, 2.2.1 and 2.2.2, which summarise the history of the BGP in the context of previous incidents and how attacks could happen. The explanation of how attacks could happen is based on the analysis of the 2008 YouTube Pakistan incident, which is going to be used as the case study for clarifying the occurrence of IP prefix hijacking. The aim of this section is to give an overview of how IP prefix hijacking occurs and where the issue lies within BGP updates.

2.2.1 History of IP prefix hijacking

This section briefly discusses the history of BGP attacks during the past two decades. Generally, attacks can occur in different parts of BGP messages. However, this research focuses on hijacks that can take place by exploiting update message flaws, such as in IP prefix hijacking. There have been several IP prefix hijacking events since 2000, but this section expands on the six worst Internet routing attacks from this period, as mentioned in [18]. The organisations subjected to attack as well as corresponding dates are explained as follows:

1 Yahoo, May 2004

DataOne (a Malaysian ISP) tried to hijack Yahoo's Santa Clara data-centre prefix in May 2004. According to network security experts, the incident was malicious [18].

2 TTNet took over the Internet, December 2004

On December 24, 2004, the Turkish TTNet ISP took over the Internet; the full table of Internet routes was sent by TTNet, which claimed that it had the best path for routing everything on the Internet, according to BGP expert Renesys. As a result, all traffic from different sites, such as Yahoo, Microsoft, Amazon and CNN, shifted to TTNet [18].

3 Con Edison hijacked a large portion of prefixes, January 2006

The normal way that ISPs route traffic to their larger customers is to wait for other customers to announce networks; then, the ISPs just propagate the announcement. However, at 05:05:33 UTC on January 22, 2006, Con Edison announced a number of prefixes pretending they are owned by their customers. A large number of announcements/propagations were made over the following several minutes, indicating the instability of the routers and that something wrong had happened. The routers continued to accept announcements until 05:22:29 UTC, when the networks started moving back to their real owners. However, At 8:23:12 UTC, Verio (aka NTT America, AS2914) started accepting some of the same fake routes that other affected routers were already spreading [25].

4 Pakistan blocks YouTube, February 2008

YouTube blocking is one of the most well-known IP prefix attacks occurred in 2008 by Pakistan Telecom. On Friday, February 22, 2008, Pakistan Telecom received an order to prevent people from accessing some videos on YouTube. Therefore, it impersonated the IP prefix and blocked YouTube for several hours. Pakistan Telecom is managed by the APNIC (Asia-Pacific Network Information Centre), while the victim is managed by the ARIN (American Registry for Internet Numbers), which means they are in two different regions. The YouTube website was completely inaccessible on Sunday, February 25, 2008, from 10:48 to 12:51 UTC [26]. On February 26, 2008, the bogus advertisement was withdrawn at the request of the government, but the ban was left in place to prevent Pakistanis from accessing YouTube [27]. As a result, the fake route was removed from all routers' routing tables.

5 Chinese ISP hijacked many different ASes, April 2010

AS23724 (CHINANET-IDC-BJ-AP IDC) is one of the data centres operated by China Telecom. Normally, AS23724 only originates 40 prefixes; however, on April 8, 2010, the ISP originated about 37,000 unique prefixes in 15-minute. Popular websites like dell.com, cnn.com, www.amazon.de, www.rapidshare.com and www.geocities.jp were affected. In addition, a large number of networks impacted Chinese websites belonging to Chinese Telecom, including www.joy.cn, www.pconline.com.cn, www.huanqiu.com, www.tianya.cn and www.chinaz.com. The incident was detected globally in The Netherlands, the UK, Russia, Italy, Sweden, the US, Japan and Brazil. This highlights the impact of hijacking on networks over a large area on the Internet [28].

6 BGP hijacking for monetary gain, August 2014

A message was sent by a user named ‘Caution’ in the bitcointalk.org forum which stated that suspicious activity was occurring on mining systems connected to the wafflepool.com mining pool. Some users in this forum and other cryptocurrency forums observed similar activity: mining systems unnoticeably redirected to an unknown IP address that answered with the Stratum protocol. Once connected to this IP address, miners continued receiving work but no longer received block rewards for their mining efforts. Hijackers exploited miners‘ hashing power by redirecting legitimate mining traffic allocated for well-known pools to a malicious server masquerading as the legitimate pool. Normally, miners continuously connect to a legitimate pool for tasks. However, in this case, when miners tried to connect to the

legitimate pool, a new BGP route directed their traffic to a pool belonging to the hijackers. Miners noticed that something wrong had happened to their transactions; therefore, the hijackers avoided the already hijacked traffic. Instead, they convinced the miners to connect to different malicious pools other than the first suspicious pool, which had already been hijacked. Miners redirected to the hijackers' pool continued to see tasks and perform work, but were not compensated; while miners who were not redirected remained unaffected. The hijacking continued for months as the hijackers repeated the process in short bursts [29].

From the above listed examples, it is clear that big organisations cannot protect themselves from the effectiveness of IP prefix hijacking that are performed by untrusted organisations because the connection among routers is global. For hijacked routers with many neighbours, their best path connection to direct and indirect neighbours is affected more than routers with fewer neighbours, as the number of withdrawing or changing routes is too high. However, organisations do not detect the hijacking because of a lack of security among the routers. If the organisations try to stop hijacking events occurring via direct neighbours by filtering incoming announcement routes, it would become very difficult to avoid hijacking from indirect neighbours, as routers with filtering do not check ASPATH lists if they have untrusted routers in between.

2.2.2 Process of IP prefix hijacking

This section discusses the process view of IP prefix hijacking and its impact on the end user. The purpose of the discussion is to help trace the main reasons behind the occurrence of IP prefix hijacking without victims' knowledge and then to identify the appropriate data needed

to disclose this attack. Along with studying the BGP's raw data during the building of a proposed system to secure or detect IP prefix hijacking, this section will also give a useful demonstration of the most appropriate methodology for solving IP prefix hijacking.

With respect to providing AS numbers and IP prefixes to routers, the IANA (Internet Assigned Numbers Authority) distributes IP prefixes and ASNs to five RIRs – ARIN, RIPE NCC, APNIC, LACNIC and AfriNIC – and other large organisations, which in turn distribute these IP prefixes and ASNs to smaller organisations. Afterwards, any router having these two data (IP prefixes and ASNs) can join the Internet and start routing information to its neighbours. In the beginning, each two directly connected routers establish a BGP session and trust each other to send and receive updates. However, any router on the Internet can be configured badly, either maliciously or due to human error, and impersonate other neighbours' IP prefixes. In normal operations, any router linked to the Internet and speaking the BGP language has three tasks – to announce, propagate or withdraw routes – to update other routers' routing tables through the sending and receiving of update packets from direct or indirect neighbours. The difference between announcing and propagating routes is that announcers who announce routes own them, while propagators only propagate the routes to their neighbours to inform them that the announcers can take them to a specific network.

After giving brief information about the needed data (e.g., AS numbers and IP prefixes) and the normal work of routers, the process by which IP prefix hijacking can occur without victims' awareness will be explained hypothetically as in Figure 2.1. This figure consists of seven ASes: The edge router (RTA) in AS100 represents the real owner (the announcer) of the prefix 1.1.1.0/24, while the router (RTD) in AS300 aims to hijack 1.1.1.0/24; the remaining edge routers, AS400, AS200, AS600, AS500 and AS700, work as propagators to

1.1.1.0/24 which anticipate the spreading of a bogus route. The edge router (RTG) in AS700 is a propagator but is in a different region (e.g., ARIN), while the other six ASes are in one region (e.g., APNIC). Edge router RTD in AS300 is the hijacker, while edge router RTE is a propagator in the same autonomous system. Every router that speaks the BGP language and is connected to other routers in a different autonomous system is called an edge router.

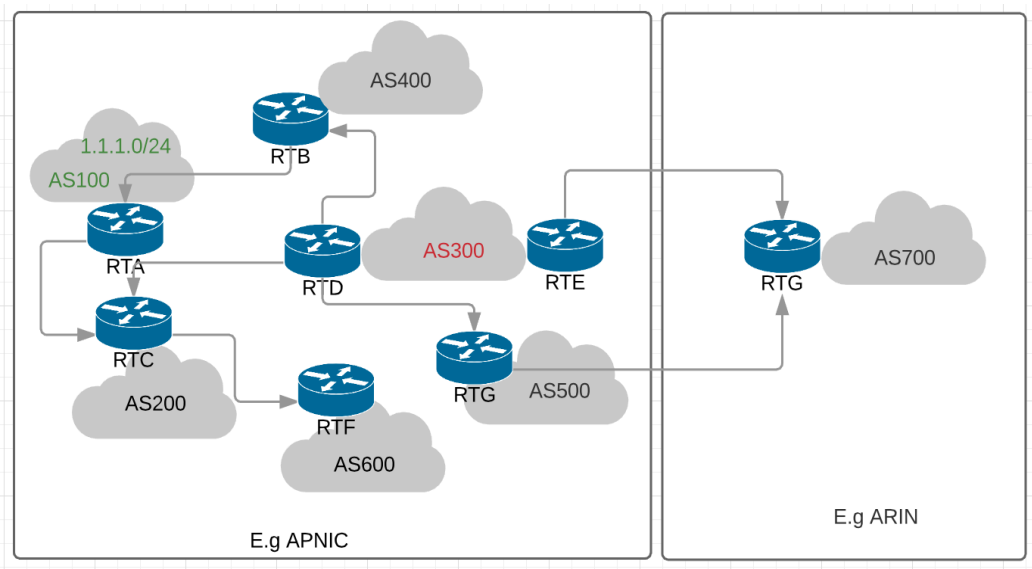


Figure 2.1 Prefix hijacking process

Edge router RTA in AS100 announces the IP prefix 1.1.1.0/24 to router RTC that is located in AS200, which, in turn, propagates it to AS600. There is no direct connection between AS300 and AS100. AS300 announces the same IP prefix to AS200 and AS500 either before or after AS100. Although both AS100 and AS300 announce the same IP prefix 1.1.1.0/24 to AS200, AS200 cannot detect the hijacking; however, it will probably spread it out to AS600 based on the filtering policy among neighbours. In addition, AS300 announces the same prefix, which it does not originally own, to AS500, then further to other national or international ASes. Moreover, AS300 announces IP prefix 1.1.1.0/24 directly to AS400,

which would perhaps announce it to the real origin AS (AS100). However, the real owner will not detect that it had been hijacked because the BGP lacks origin authentication. In this case, some edge routers would assume that AS300 could forward traffic for 1.1.1.0/24 via a route with a lower cost than the one offered by AS100. Consequently, edge routers would redirect their routes and use AS300. If the case is applied to extremely popular organisations like YouTube, Google and Yahoo, denial of service will have an immediate impact on end users.

2.3 Securing the BGP

This section discusses a number of BGP security solutions that have been proposed to prevent IP prefix hijacking in BGP version 4, such as sBGP, soBGP, psBGP and RPKI (Resource Public Key Infrastructure). These solutions were proposed theoretically and are based on nearly similar mechanisms to authenticate BGP update messages and authorise ASes that can advertise a specific IP prefix. Each solution is described according to the security techniques used and methodologies followed for processing and structuring data. The solutions are evaluated based on different, important features of the security system (solution), such as computational complexity, deployment susceptibility and quality.

2.3.1 secure Border Gateway Protocol (sBGP)

sBGP is a protocol proposed to satisfy BGP security requirements. This protocol uses common security mechanisms, such as the PKI, attestations and IPsec to prevent illegal operations among routers. The main purpose of the protocol is to prevent IP prefix hijacking

through the authentication of BGP update messages and the authorisation of ASes to deliver a particular IP prefix, as well as by verifying that received messages are sent by owners.

2.3.1.1 sBGP mechanism

sBGP uses three techniques to secure itself and prevent serious attacks like IP prefix hijacking. First, sBGP uses PKIs as its primary technique; this technique is based on cryptographic key management, whereby private keys and public keys are managed by a trusted root certificate authority such as the IANA or ICANN (Internet Corporation for Assigned Names and Numbers). These keys can be delegated to RIRs in order to distribute them over the ASes. During the sending of a BGP packet, the ASes use private keys to sign update messages and, upon receiving these messages, routers verify them using the public keys of the senders [13], [19].

The second technique that sBGP depends on is called attestations. This approach is used so that sBGP can encapsulate authorisation information in an UPDATE message; then, the information is signed digitally to ensure the authenticity and integrity of data provided in the update message. By using attestations, the sBGP allows each AS to check the authority of advertising an IP prefix to other ASes along the path, and to verify that the advertising AS was authorised by the owners themselves [19]. For example, in route (100,200,300,400), AS400, which is the owner, needs to give authority to all ASes (300,200,100) in the path to propagate a specific IP prefix. Similarly, each AS in the route must give authority to the next one in order to propagate the same IP prefix.

The last security technique utilised in this solution is IPSec; this protocol can secure point-to-point communication between speakers. In other words, it works on an OPEN message level not on an update message level. IPSec includes many services related to security which could be applied to the BGP, such as access control service, connectionless data integrity service, data origin authentication service and data confidentiality service. The ability to manipulate the order of AS numbers in the ASPATH attribute is called a modification and deletion attack. This manipulation can happen either by withdrawing a valid route or intercepting the ASPATH. Attackers could apply these attacks by deleting particular ASNs or modifying specific paths to redirect packets, or by causing DoS attacks by making a specific network unreachable [13].

2.3.1.2 Benefits and limitations

In terms of advantages, the sBGP is considered a very concrete solution because it has the ability to sign, verify and validate ASes and IP prefixes dynamically and simultaneously. However, sBGP routers must give authenticity to their neighbours to propagate a specific IP prefix, but the sBGP routers cannot guarantee that trusted neighbours do not themselves claim the ownership of the IP prefix. As a result, preventing IP prefix hijacking would be very difficult in the sBGP. The protocol depends on a hierarchal structure of multiple levels of trusted certificate authorities for signing ASNs and has a complicated management process

Due to the need for cooperation among routers to deliver update packets, any trusted routers can act up and falsify the ASPATH attribute values. It is true that their effectiveness would be limited because only trusted routers can falsify the ASPATH, but hijacking can still be achieved. Moreover, attestations are used in the sBGP to verify the ability of ASes to

propagate IP prefixes by limited ASes [19]. Similar to this aim is already available in BGP community filtering commands which allow using no-export attribute to propagate IP prefixes by specific ASes [30]. As pointed out by Aiello et al. [31], it is exceptionally difficult to approximate an IP address delegation graph for the Internet. Therefore, it might be impossible to build centralised PKI mirroring such as a complex and unknown delegation structure. Although the sBGP includes all IPSec services, these services are not fundamental requirements for inter-domain routing itself.

2.3.2 secure origin Border Gateway Protocol (soBGP)

The soBGP (secure origin Border Gateway Protocol) aims to secure the traditional BGP, specifically by preventing IP prefix hijacking. The protocol secures the BGP based on four certificates: AS Policy Certificates (ASPolicyCerts), Authorisation Certificates (AuthCerts), Entity Certificates (EntityCerts) and Prefix Policy Certificates (PrefixPolicyCerts). ASPolicyCerts are used to assert routing policy for an AS, while AuthCerts binds ASes to their IP prefixes to assert the authorisation of an AS to advertise a specific prefix. EntityCerts assert mapping between ASes and their public keys, while PrefixPolicyCerts are responsible for mapping ASes to one or more prefixes [15]. All of these certificates have a similar mechanism but work on different entities (e.g., AS number, AS policy and prefix policy).

The soBGP is based on three common security techniques: authentication, verification and the web-trust-model. Authentication is used to authenticate policies in the routing system, such as AS policy and prefix policy, while verification is used to verify them. In the authentication phase, the soBGP uses the web-trust-model, which is a technique for authenticating ASes and linking them to their public keys. In the verification phase, the

soBGP uses AS public key signatures so they can validate each other [13]. Section 2.3.2.1 will detail the authentication and verification mechanisms.

2.3.2.1 soBGP mechanism

In the authentication phase, as explained in section 2.3.2, EntityCert certificates are generated by the CA (Certificate Authority), which bind AS numbers with their public keys. These certificates are signed digitally by Tier-1 ISPs or well-known authentication service providers such as VeriSign. To achieve trust among routers, a small number of 'root public key certificates' are distributed among them using out-of-band mechanisms. An AS with a trusted AS public key certificate, already signed by a trusted CA, might use its private key to issue and sign a further public key certificate (EntityCert) for another AS in a hierarchical structure that will naturally form a web-of-trust model. The other three certificates mentioned in section 2.3.2 are issued among ASes with no involvement of certificate authorities. Afterwards, these three certificates are distributed over BGP routers while routing information in the update messages [13].

In the verification phase, when a router receives an update message, it checks both the advertisement authority of each BGP router through the sender's public key signature available in the EntityCert and the IP prefix ownership via the signatures in the AuthCerts. The other two certificates, ASPolicyCerts and PrefixPolicyCerts, are used by routers to check the policy of ASes and IP prefixes in order to avoid false positives that routing policies, such as address aggregation and AS confederation, could result in during the verification of an IP prefix ownership. For example, in a case where router A carries four certificates and wants to exchange routing information with router B, router B will not trust router A until it already

has all four certificates' public keys of router A; it then checks its signatures before the announcement is used and stored in the routing table or propagated to another neighbour [19].

2.3.2.2 Benefits and limitations

In terms of authentication, the soBGP uses four certificates to authenticate ASes and IP prefix routing polices which lead to an increase in accuracy and a reduction in false positives. The web-of-trust model is used by the soBGP and has strong proponents for authenticating user public keys within the technical PKI community [39]. With respect to IP prefix ownership verification, the soBGP makes use of a strictly hierarchical structure, and prefix delegation structures might be simplified in the soBGP by using ASes instead of organisations.

Using the web-of-trust model with the soBGP is suitable for authenticating AS public keys, which are identified by AS numbers strictly controlled by the IANA; thus, it is questionable whether any entity other than the IANA should be trusted to sign AS public key certificates. Kranakis et al. suggest that the soBGP, like the sBGP, also faces difficulties tracing changes of IP address ownership in a strictly hierarchical way [13]. In addition, it is not clear whether it is practical to use hierarchical structure since IP addresses are usually delegated to organisations, not to ASes [2]. Any peer could join the Internet, work as a trusted entity, and then misbehave; in this case, all mechanisms will be ineffective. Finally, the soBGP does not have a mechanism for identifying invalid certificates (expired certificates); therefore, this makes the soBGP less secure.

2.3.3 pretty secure Border Gateway Protocol (psBGP)

The pretty secure Border Gateway Protocol (psBGP) is a combination of practical solutions using the best features of the sBGP and soBGP which aims to secure the BGP [32]. The psBGP consists of two models: the centralised trust model and the decentralised trust model. The protocol uses the centralised trust model for authenticating AS numbers and BGP speakers, and the decentralised trust model for verifying IP prefix ownership and ASPATH [33]. The mechanism of this protocol will be explained in the context of five main security goals: The first and second goals are related to data origin authentication for ASNs and BGP speakers, while the third goal is concerned with data integrity that does not add any additional security to the BGP because it is achieved implicitly when the BGP establishes TCP peering sessions with neighbours. In other words, the TCP already has IPSec, which performs the same task; therefore, this section will not talk about the data integrity goal. The fourth and fifth goals focus on a way to verify prefix origination and ASPATH during the exchange of routing information [32]. The objective of the psBGP is to explore alternative policies and trade-offs to provide an acceptable balance between practicality and security.

2.3.3.1 psBGP mechanism

It would be useful to start by explaining the architecture of the two trusted models on which the psBGP is based for authenticating ASNs and BGP speakers and asserting ownership of IP prefixes and ASPATH; then the mechanism for and verification of these data to prevent IP prefix hijacking could be demonstrated. As mentioned above, data integrity will not be discussed in this subsection because it does not add any direct security contribution to the components of inter-domain protocols.

First, the centralised trust model uses RIRs as the root of trusted certificate authorities like big ISPs. In general, RIRs generate public key certificates and sign them for association with ASNs. When an organisation applies for an AS number, the RIR will bind the AS number to a certificate to issue another certificate called the ASNumCert. ISPs follow the same procedure with customers applying for the ASNumCert. This distributes the effort of providing ASNumCerts among AS number providers (e.g., RIRs and ISPs). ASes also need to keep their private keys, which correspond to their issued public keys, to prove the authenticity of their specific ASNumCert to the RIR when they need to modify the certificate or for any other reason. In terms of authenticating BGP speakers, the psBGP uses the same model. An AS with a certified ASNumCert issues an operational public key certificate shared by all BGP speakers within the AS – namely, the SpeakerCert. BGP speakers need to have the private keys that have already been issued and which correspond to the public keys of the ASes with ASNumCerts to sign the SpeakerCert [32].

To verify AS numbers, the psBGP assumes that BGP peers on the Internet have already been provided with all neighbours' ASNumCerts through out-of-band mechanisms, taking into account the fact that the ASNumCert is revoked when the corresponding AS number is not used or reassigned to another organisation. While changing BGP update messages between BGP peers, each AS sends its ASNumCert, which has been issued and granted by a RIR, in the BGP update message so peers can verify its AS number. Peers in turn verify every received announcement based on the ASNumCert attached to the BGP update message. If the verification is successful, peers accept the announcement; otherwise, it is rejected.

To verify the BGP, speakers use SpeakerCerts already issued in the authentication phase. SpeakerCerts are distributed among BGP speakers upon sending and receiving update

messages. SpeakerCerts are used for establishing secure connections with peers and for signing BGP messages. Assuming that BGP speakers have their peers' public keys, when a BGP speaker receives an update message from its peer, it will use its public key to verify the identification based on the SpeakerCert, which includes the signature of the peer.

Second, the psBGP uses a decentralised trust model for verifying the propriety of IP prefix ownership and ASPATH. The architecture of this model is based on lists of ASNs bound with their IP prefixes. In other words, each AS creates a prefix assertion list (PAL) that consists of a number of bindings of AS numbers and prefixes. The first assertion in the PAL is allocated for the AS itself, and the other assertions belong to the peering ASes where the assertions (the endorsements of an AS's peers) are ordered based on the ASNs [13].

In terms of verifying the origination of a specific prefix to an AS, the AS's peers utilise the architecture of the decentralised trusted model. Generally, each peer on the Internet needs to have prior knowledge of some level of due diligence offline to determine what IP prefixes are delegated to each of its peers [32]. Based on that delegation knowledge, peers will have the origination of prefix endorsements (assertions) of an AS. When the AS wants to announce a specific prefix, peers on the other side will check the consistency of peers' assertions in the PALs to verify its ownership to the prefix [13]. If at least one peer asserts that the AS owns the prefix, the BGP packet will be accepted; otherwise, it will be rejected.

ASPATH is a BGP attribute that consists of a list of AS numbers and is always sent with update messages. To verify ASPATHs, the psBGP uses a bit vector data structure in the PAL to make the operation very quick. The PAL structure in ASPATH verification is devolved to take triple format, {prefix, [P1, P2, P3, ..., Pi], Vi[ni]}, instead of duple format,{fi, [P1, P2,

P3, ..., Pi]}, where fi is endorsed prefixes, Pi is peers of an AS, Vi is a bit vector, and ni is the corresponding length. A bit vector is a data structure array that compactly stores bits. psBGP speakers use digital signatures to sign the new structure of PALs and, based on the the signature, verify each other. For example, [P1, P2, P3] represents a list of an ASPATH; however, P3 will not accept an update message from P2 until it has the digital signatures of {f1, [P1], v1[n1]}P1, {f1, [P1, P2],v2[n2]}P2 from P2 [32].

2.3.3.2 Benefits and limitations

Compared to the sBGP, the psBGP signs ASN certificates received from RIRs or trusted authorities directly, consequently reducing the certificate management burden, while the sBGP depends on a hierarchal structure of multiple levels of trusted certificate authorities for signing ASNs and has a complicated management process. Another advantage of the psBGP solution is that it can address uncoordinated, misconfigured and malicious BGP routers [33]. The psBGP is able to distribute the difficult task of tracing IP address ownership across all ASes on the Internet by making each AS verify its peering ASes based on the PALs [33]. The third advantage of the psBGP is that it includes a method for describing IP prefix engineering such as IP address aggregation [32], which increases the accuracy of the protocol and reduces false positives during the verification of prefix origination and ASPATH.

The psBGP also has two serious drawbacks with regard to verifying the ownership of IP prefixes when it uses the decentralised trusted model. First, some ASes could have only one neighbour AS, which would in turn mean that their PALs would only have one prefix assertion. In this case, peers of these ASes would not be able to check or compare the consistency of the assertions because they would receive a PAL with only one assertion from

these kinds of ASes. Second, some ASes leave their entry in the PAL empty or null, which means no endorsement was given by those ASes to their peers [13]. In other words, the psBGP would not be able to stop ASes that have null assertion entries from achieving IP prefix hijacking or affecting the robustness of the protocol.

2.3.4 Preventing IP prefix hijacking using the RPKI

The RPKI (Resource Public Key Infrastructure) is a framework proposed by the IETF (Internet Engineering Task Force) to secure the inter-domain routing system. This framework uses three security techniques to secure the BGP and prevent IP prefix hijacking: the PKI, signatures and positive attestations [12]. The PKI is used to allow ASes to generate their own private and public keys, while signatures are allocated to use these keys for signing ASes and IP prefixes and for verifying routes. Positive attestations are used to allow a RP to validate ROAs within the RPKI. Substantively, the RPKI tries to find a way to make the hierarchical structure delegation of the prefixes and ASNs more secure [34]. The main purpose of this technique is to address misconfigurations and hijacking by identifying the real owner of a particular IP address. The following subsection discusses the mechanism of the RPKI and how to protect a specific address space.

2.3.4.1 RPKI mechanism

The RPKI is an approach for building formally verifiable documents, including IP addresses and their origin ASes. In other words, the RPKI uses the special structure of linking origin ASes and their IP prefixes and puts them in special documents for route validation; these

documents are called ROAs (Route Origin Authorisations) [24]. The aim of ROAs is to give authorisation to the ASes to announce their IP prefixes [11].

In the beginning, ASes generate their own private and public keys to sign ROAs and validate routes, respectively. The signed objects are saved in an RPKI cache and must be refreshed periodically by BGP speakers [24]. Each AS uses its private key and digitally signs the linked AS with its IP prefixes and publishes the public key to be used for verifying its signature. On the other side, ASes that receive the announcement will wait for a RP (replying party) to validate the routes. A third party will use the announcer's public key to check if the AS is authorised to originate the IP prefixes [24].

The RP assumes that it has full access to the ROAs to check the validity of a route. Generally, the RP is a server which provides access to a secure software application. This server is set logically between routers to check the origination of the routes in the ROAs using their origin Ases' public keys. If a route does not pass the validation phase, it will be interpreted by the RP as 'invalid' and the router will not be able to originate the desirable IP prefixes [34][12].

The RPKI proposes that routers accept and refuse routes based on three validation states: not found, valid and invalid. Not found means the checked route prefix does not exist among ROAs, while valid means at least one ROA matches the route prefix; finally, invalid means that at least one of the ROAs matches the route prefix, but in the validation no one matches it [24].

2.3.4.2 Benefits and limitations

In the context of router functionality, IP address aggregation and AS confederation operations are used for specific engineering purposes, such as announcing an IP prefix on behalf of other ASes. However, these operations could make following the origination of specific IP prefixes very complicated. In other words, some ASes do not or might forget to describe all router policies regarding engineering IP addresses, which could lead to many false positives.

Another drawback of the RPKI is that some attackers can inject multiple valid ROAs. Since the third party (e.g., RP) matches received announcement routes with the objects in the ROAs but does not check if there is more than one AS claiming the same IP prefixes in the RPKI, the injected valid ROAs will make the RP exceed the validation.

Wählisch et al. built a system in [12] to prevent IP prefix hijacking that uses the RPKI to detect IP prefix hijacking. They tried to determine the reasons that making the announcements invalid to pass the route validation to compute false negative and false positive. In other words, they could not compute the false positives and negatives accurately. However, the authors indicate that hijacking often happens because of unallocated address spaces, which were not addressed in the RPKI.

The RPKI uses three validation states to accept or refuse announcements, valid or invalid or not found [24], but the problem with validation states is that they cannot determine the main reason for invalid announcements. If an operator misses more specific prefixes to add to ROAs, the system will fail to prevent hijackings [12]. In other words, the RPKI requires that all ASes provide their associated origin ASes with IP prefixes and describe them accurately

in ROAs. If some legitimate ASes do not provide their ROAs to the RPKI, then the RPKI framework will not be able to know the accurate cases of the announced route prefix.

2.3.5 Limitations of existing prevention solutions

Prevention hijacking solutions to authenticate the identity of the routers are usually based on security techniques such as the PKI, attestations and digital signatures. These techniques are very strong and useful for individual tasks such as accessing an account in a specific server but not for exchanging BGP packets among routers because of the large size of the packets being exchanged. In addition, prevention solutions are not yet deployable and are subject to some false positives, as explained in section 2.3.4 for the RPKI. Some of the other proposed prevention solutions were mentioned in [9], [35], [36]. Moreover, if an attack takes place, these solutions, due to the fact that they are based on authentication, would not be able to detect it at all. As a result, a more effective approach would be to use anomaly detection techniques, which will be explained in section 2.4. Although prevention techniques are very important, detection techniques are more feasible because they can trace the hijacking continuously and keep secured systems up to date. However, security solutions do not achieve these features; if an attack passes authentication, such solutions will not be able to detect or remove it.

2.4 IP prefix hijacking detection

This section discusses a different way to secure the BGP by detecting IP prefix hijacking before it spreads out. This is a different technique from the one discussed in section 2.3. Generally, there are many anomaly detection techniques regularly used to detect anomalies

that threaten networks, such as rule-based, finite state machines, statistical analysis and pattern matching [37]. However, this section will be limited to five examples of anomaly detection techniques that have already been applied to the BGP in order to reflect their effectiveness for detecting IP prefix hijacking. These examples should cover most of the methods that have been used to detect hijacking in the BGP. Other solutions will be omitted because they are either not directly related to detecting IP prefix hijacking but rather more general events such as worms, blackouts and misconfiguration [38], or they follow the same detection methods. Discussing the resources of the previously proposed detection solutions based on the detection of IP prefix hijacking would be a good place to begin, followed by an explanation of the detection methods or solution mechanisms.

Previous, different solutions are summarised in five detection approaches: (1) rule-based, which concerns information that can be used to explore and check a specific issue in the BGP such as unallocated IP prefixes (found in the IANA and RIRs) that can be exploited to cause disruption in networks or MOAS (Multiple Origin Autonomous System) conflicts, which make data flow in ASes invalid and prefix hijacking achievable [16]; (2) monitoring-based, which focuses on tracking network reachability among transit networks by using lightweight prefix-owner-based and active probing to detect IP prefix hijacking [20]; (3) origin changes monitoring-based, which is concerning with following up on the changes of the origins of a specific prefix [39]; (4) historical-based, which applies to how to construct a set of recently seen data (prefix, origin AS, time) in a historical window [40]; and finally (5) multiple events monitoring-based, which concentrates on the changes that occur to BGP routing during the occurrence of an event such as spam and suspicious MOASs [38]. These five detection techniques will be discussed from four angles: detection mechanisms, experiments used, detection results, and evaluation of the solutions.

2.4.1 Accuracy and challenges of accessing BGP resources

Selecting accurate data resources is an important factor for solutions trying to secure the BGP; therefore, this section is allocated towards investigating the accuracy and availability of the data collection resources used by previous solutions before the discussion of detection methods or mechanisms begins. First, BGP-related resources are divided into two categories from which IP prefix hijacking detection solutions can collect their data: registry-based and trace-based. The trace-based category represents active routing information, while the registry-based category concerns registered routing information. Routing information can be obtained from routing tables and update messages, while registered routing information is included in RIRs and IRRs (Internet Routing Registries). Routing tables and update messages are available in the BGP routers themselves. However, registered routing data can be obtained from five different global registries: the African Network Information Center (AfriNIC), the American Registry for Internet Numbers (ARIN), the Asia-Pacific Network Information Centre (APNIC), the Latin America and Caribbean Network Information Centre (LACNIC) and Réseaux IP Européens Network Coordination Centre (RIPE NCC) [41]. The investigation of these resources is needed to evaluate previous work and choose the most appropriate information to be used with the proposed detection methods discussed in Chapter Chapter : 3, Chapter : 4 and Chapter : 5.

2.4.1.1 Routing tables and BGP update messages

BGP update message information is considered the main generator of information stored in routing tables. BGP update messages contain seven types of information shared in all BGP packets: the time and date the BGP packet was sent at and on, message type, sending from

and receiving to routers, and source and destination ASes. BGP update messages have 14 discretionary and mandatory BGP attributes. Based on this information, BGP routers build their routing tables from exchanged routing information, such as ASes, prefixes, router IDs and so on. Routing tables and update messages can be collected from Route Views, which is available on [21].

In terms of accuracy, update messages are very accurate and are considered real-time data. Routing tables contain voluminous routing entries, which leads to great cost during the checking of routing tables. In the meantime, the routing table is relatively stable such that it is not worth checking it repeatedly. It also would be very difficult to detect short-lived anomalies (e.g., prefix hijacking). In the context of challenges, some organisations are not willing to provide their routing tables to researchers because of privacy considerations [16]. Therefore, routing tables are not a good choice for detecting IP prefix hijacking.

2.4.1.2 Regional Internet Registries and Internet Routing Registries

RIRs are online databases that are typically used to retrieve specific information such as AS numbers, IP prefixes and organisation names; while IRRs, such as the Reseaux Internet Protocol Europeans-Routing Information Service (RIPE-RIS), are looking glass databases that extract their registered information from RIRs and make them available to the research community. Both databases can be used for avoiding problematic issues between ISPs and globally to help network operators debug routing tables and configure routers properly. In addition, RIRs and IRRs can be used as mechanisms for allowing the validation of BGP announcement message content or mapping an origin AS number to a list of IP prefixes [42].

Nemecis (NEtwork ManagEment and Configuration System), which is a tool used in [43] to evaluate the accuracy of registered routing information resources, has also been used with IP prefix detection solutions, such as IRRs. As in [44], Nemecis verifies the existence and consistency of ASes and prefix registration objects against BGP updates by matching various attributes such as organisation, maintainer, email handle, etc. Because the asynchronous changing among active update messages and information is stored in the IRRs, the algorithm in principle generates alerts if the checks fail, i.e., when there is a lack of a full or partial consistency check. A full consistency check, for example, occurs when the route object is consistent with the prefix and the autonomous number; whereas a partial consistency check occurs when the route object is consistent with only the prefix or the autonomous number. Based on the results of Nemecis, the authors in [43] claim that IRRs are not accurate.

In terms of challenges, RIRs cannot be linked to research in a programmable way like IRRs; both resources need to be concurrent and up to date with any detection methods based on them; otherwise, the methods could be subject to false positives.

2.4.1.3 Comparing between registered and routing information

Even though IRRs are trusted sources, according to [43], they might be inaccurate based on [44]. Therefore, IRRs are not reliable for use as data resources with IP prefix hijacking detection solutions. IRRs store irregularly updated registered routing information, but BGP security tools need to debug and validate them. The contents of BGP update messages are changed both periodically and continuously. Thus, BGP update messages should be combined with registered-based information to detect IP prefix hijacking.

2.4.2 Using rule-based and packet-based approaches with prefix relationships

Rule-based approach can be used in problem solving to detect unexpected events. This approach was mentioned in [37], [16], [45]–[49] as one of the detection methods that can be used to detect anomalies in communication networks, including IP prefix hijacking. The packet-based approach is a method of checking real-time data received from routers. Wang et al. proposed in [12] a novel model that combines packet-based and rule-based approaches. This model obtains routing information from various AS edge routers so that it can come up with more effective detection. The rule-based approach is used to check the contents of routing packets through the RIS (Routing Information Service) or RIDB (Routing Information Database) [50] (e.g., Internet Routing Registries), which include the registered routing information of organisations. The RIS project is a service that provides a collection of global routing information.

This model consists of four main components: (1) The RIS Server, where the model collects registered routing information from; (2) the Data Collector, which is a PC responsible for receiving routing packets from different ASes; (3) the RIS Adapter, which is used to gather registered routing information from different sources and then put it in corresponding and structured databases; and finally (4) the Detection Server, which is allocated to detect anomalies in the BGP based on the seven suggested rules [16]. Figure **2.2** shows the architecture of these four main components.

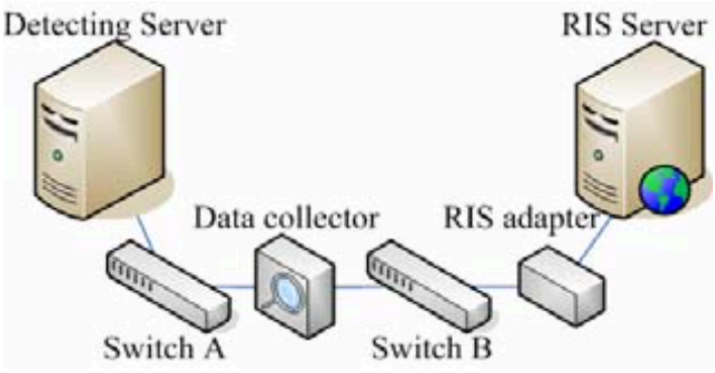


Figure 2.2 The architecture of the model [16]

The model uses six rules to detect different anomalies in the BGP, including IP prefix hijacking. These rules are summarised as Prefixes of reserved and unallocated IP blocks, MOAS conflicts, private and unallocated ASes in ASPATH, repeated ASes in ASPATH, AS loops in ASPATH, ASPATH's violation against hierarchy, and finally illegal links to foreign ASes in ASPATH. The first and third rules were discussed in detail in [4]. From the proposed rules, the combination model tries to solve seven issues, along with IP prefix hijacking, which face the BGP. It is apparent from Table **2-1** that the rules are taken from two fields, ASPATH and IP prefix. The rules are based on ASPATH and are allocated to track data flow of invalid ASes (e.g., traffic leaks, unexpected ASes to appear in ASPATH, low efficiency of the Internet-looped packets, ASes in lower degree should not forward traffic for more than one high-degree ASes and AS links between different countries should not be arbitrary and should be authorised) while the one rule related to the IP prefix is specified for detecting disruptions in the network. However, this section will only focus on how to use the MOAS conflicts rule because it is considered related to IP prefix hijacking. MOAS means that two different ASes announce either one super-IP prefix or the sub-IP prefix of the other. This operation reflects the signature of IP prefix hijacking. By the end, the combination model will

use the MOAS conflicts rule, exploiting the observation of AS prefix relationships to detect IP prefix hijacking. The subsections below discuss the use of the detection model in detail.

NO	Rules	Data category
1	Prefix of reserved and unallocated IP blocks	PREFIX
2	MOAS conflicts	ASPATH and PREFIX
3	Private and unallocated AS in ASPATH	ASPATH
4	Repeated ASes in ASPATH	ASPATH
5	AS loop in ASPATH	ASPATH
6	ASPATH's violation against hierarchy	ASPATH

Table 2-1 Detection rules of the model

2.4.2.1 Data collection

The Data Collector is a PC that collects BGP routing information from different ISPs or ASes. The Data Collector needs to be connected to BGP routers directly and acts as a speaker using routing emulation software such as Zebra [51]. Zebra is installed on the PC and establishes a dummy BGP session with real neighbours to collect basic data from them, such as AS numbers, prefixes of organisations and allocated and reserved IP blocks, and routing control information. The PC has many different data collectors so it can collect a large volume of routing information from different ASes.

The RIS Server is a machine in which registered routing information is stored. The registered information is basically stored in a RIDB (Routing Information Database) and is available to the public, including network operators, to debug misconfigurations that could happen by

mistake; it is also available for community research. However, the model can benefit from this information for verifying the proposed anomaly detection rules. Since the routing information of the proposed anomaly detection rules are not available in one source, the information in the RIDB is taken from different sources. For example, AS information is obtained from statistical reports from the CAIDA Corporation. However, prefixes of organisations are collected from implemented RIR Whois servers, which only provide prefixes in their own region; but with the collaboration of the Completewhois server, prefixes can be provided for different regions. Moreover, allocated and reserved IP blocks are taken from the IANA and, finally, information about specific ASes is obtained from AS administrators. Since RIDB information is taken from different sources, the model has a host, called the RIS Adapter, which is linked to the RIS Server to put gathered data in corresponding and structured databases.

2.4.2.2 Detection mechanism

In order to apply the MOAS conflicts rule, the Detection Server starts by receiving routing information, which is collected by Zebra to check update packets through the adapted registered routing information. Based on four observed relationships among IP blocks, the detection engine (Detection Server) verifies the ownership of ASes to a specific IP prefix. These relationships are summarised as any two IP blocks that might be disjointed, intersected, coincided, or subsumed within another. The first two relationships are not related to IP prefix hijacking, while the last two relationships can disclose IP prefix hijacking. Therefore, the detection engine uses coincided relationships to detect super-IP prefix hijacking, and one IP prefix is included by another relationship to detect sub-IP prefix hijacking. The detection engine checks each update packet collected via the Data Collector,

based on the adapted database. For example, if there is a newly announced prefix, the detection engine checks its relationship with the IP prefixes of all ASes in the database, which are previously prepared and adapted by the RIS Adapter; and if it coincides with or is included by other AS prefixes, the detection engine considers the announcement an IP prefix hijacking.

2.4.2.3 Experiment

The authors in [16] used the 2008 YouTube Pakistan IP prefix hijacking as a case study to evaluate the proposed model. A sample of the routing packet from the day (February 24, 2008) that the IP prefix hijacking occurred was taken, exactly between 13:07 pm and 21:19 pm UTC [52]. This sample was sent as an input to the Detection Server, which was already prepared with the detection rules, including MOAS conflicts, and the structured, adapted RIDB. The Detection Server observed that the IP prefix hijacking caused unexpected instability of the routers. As a result, many MOAS conflicts, along with four other events, were considered anomalies by the model: AS loops, private and unallocated ASes, violations of hierarchy and repeated ASes were detected in only five minutes.

2.4.2.4 Results

The rule-based and packet-based model proposes seven rules to detect different anomalies in the BGP. This section will briefly mention the final results of detecting the anomalies in the BGP, including MOAS conflicts, which affect the stability of BGP routers; these results are shown in Table **2-2**. The majority of anomalies found are related to repeated ASes, while reserved and unallocated IPes are the lowest detection because they have zero violation.

Violation of hierarchy and private and unallocated ASes are considered the second highest anomalies detected by the model. AS loops are listed as the second-to-last violation among the six anomalies that could threaten the stability of the BGP. In terms of the detection of MOAS conflicts, the model could detect 4159 cases out of the 520 173 received in five minutes.

No	Rules	Detected anomalies
1	Reserved and unallocated IPes	0
2	AS loop	47
3	Private and unallocated ASes	203
4	Violation of hierarchy	817
5	MOAS conflicts (prefix hijack)	4159
6	Repeated AS	80 287

Table 2-2 Results of anomaly detection of the rule- and packet-based models

2.4.2.5 Benefits and limitations of rule- and packet-based models

On the one hand, the authors in [16] claim that their model can achieve three main purposes: It is based on real data, the results of the model show up every five minutes, and it is effective for monitoring large-scale networks because it receives BGP packets from many different ASes. Moreover, due to collecting routing information from several ASes, Cao et al. in [16] indicate that their model is very accurate. In addition, getting routing information from different resources ensures the reliability of the Routing Information Database. Lastly, the detection engine, already installed on the RIS Server, can check unseen ASPATHs and their prefixes with the adapted database.

On the other hand, the combination of the rule-based and packet-based models has four serious limitations that have not been taken into account. First, the proposed model might be subject to false positives because it is only based on the relationship between IP blocks, yet one organisation can announce their super- and sub-prefixes with two different AS numbers. In this case, the model would consider the announcement an anomaly or a hijacking. Second, since the model (RIS Adapter) gathers information from different sources (RIDBs) and adapts them, it would have difficulty following registered routing information updates concurrently with the adaptive database. Third, the model needs to check the accuracy of the RIDBs with every announcement being checked, but doing so costs a great deal of time, which, by the end, can result in a delay of the detection.

2.4.3 Using monitoring network reachability-based approach

The iSPY is a system proposed as one of the solutions that can detect IP prefix hijacking in the BGP. This system is based on the observation of the reachability among ASes. Thus, the iSPY needs to implement a framework to monitor network reachability from transit networks to one's own specific network, which will generate a prefix-owner view. In section 2.4.3.1, the iSPY applies active probing to ASes using different IPes collected from different sources. Active probing can be performed through different network tools, such as traceroutes and TCP pings. Using traceroutes to generate a prefix-owner's view will help to build vPaths (victim paths).

The vPath represents the paths from the victim to different ASes. These paths need to be taken before and after the hijacking. The vPath will also show the size of unreachability to ASes, which will guide the iSPY to detect IP prefix hijacking. In section 2.4.3.1, an

experiment will be performed [20] involving the injection of IP prefix hijacking into three ASes located in different regions: Seattle, London and Japan. The experiment will be conducted to evaluate the methodology of using monitoring network-reachability-based approach. The last subsection will discuss the efficiency of the iSPY during the detection of IP prefix hijacking.

2.4.3.1 Data collection and detection mechanism

The iSPY collects potential live IPes from three different sources: Route Views routing tables, DNS server logs and web server logs of a university. In the first phase, the iSPY stores the IPes in a database (file) for active probing. Afterwards, the iSPY uses the traceroute tool to probe ASes on the Internet using their IPes in order to return with the reachability among these ASes through the levels of ASPATH. The iSPY also utilises ICMP pings and TCP connections at port 80 to check the liveness (reachability) of IPes and filter out their unresponsive parts. Each transit AS (can forward traffic) needs to own at least one active IP.

In the data analysis phase, the iSPY uses BGP routing tables in Route Views to generate IP-to-AS mapping to link the IPes to their ASes and put them in a database. While probing ASes through IPes in the database, the iSPY lately observes that one IP can have multiple ASes, but these cases are ignored because the infrastructure of the system was based only on monitoring the consistency of ASes reachability. In addition, the traceroute sometimes returns with *, which indicates unreachability to some ASes. Moreover, the traceroute occasionally cannot return with the AS number (or organisation name) of specific IPes. In the last two critical cases, * and no information corresponding to propping IPes, the iSPY has its own special mechanism to deal with them. If * appears between the same two AS numbers

(e.g., [1239 * 1239]), the iSPY considers the IP to belong to AS1239 and then collapses them into one AS; otherwise, it is just marked unmapped. IPes that do not have corresponding information regarding the name of the organisation are also marked unmapped. Berkeley University traced its route to the CNN organisation and published the snapshot in [53], which can give a picture of the two cases as shown in Table 2-3. More specifically, hop 7, 8 and 11 are going to be marked as unmapped hosts by the iSPY during the implementation of the AS-level.

Hops	IPes	Organisation
1	169.229.62.1	inr-daedalus-0.CS.Berkeley.EDU
2	169.229.59.225	soda-cr-1-1-soda-br-6-2
3	128.32.255.169	vlan242.inr-202-doecev.Berkeley.EDU
4	128.32.0.249	gigE6-0-0.inr-666-doecev.Berkeley.EDU
5	128.32.0.66	qsv-juniper--ucb-gw.calren2.net
6	209.247.159.109	POS1-0.hsipaccess1.SanJose1.Level3.net
7	*	?
8	64.159.1.46	?
9	209.247.9.170	pos8-0.hsa2.Atlanta2.Level3.net
10	66.185.138.33	pop2-atm-P0-2.atdn.net
11	*	?
12	66.185.136.17	pop1-atl-P4-0.atdn.net
13	64.236.16.52	www4.cnn.com

Table 2-3 Traceroute from Berkeley (169.229.62.1) to www.cnn.com (64.236.16.52) [53]

To implement the AS-level traceroute path, the iSPY maps different IPes into the same AS, as shown in Table **2-4**, based on the data available to the Route Views routing tables. This process is called resolving AS-level path to IP-level paths. Table **2-4** shows the last format that can describe the reachability of Berkeley University to CNN. Based on the cases in which the ASes are marked unmapped, hosts in hop 7, 8 and 11 are considered reachable because they are located between the same previous and next ASes. However, any cases different from the ones shown in hop 7, 8 and 11 are considered unreachable, and based on

this reachability percentage, the iSPY decides if the event is a hijacking. The format of the vPath is drawn like the one shown in Table **2-5**, which shows that there is no hijacking between Berkeley University and the CNN organisation. In section 2.4.3.2, three IP prefix hijacking will be performed onto three different organisations to evaluate prefix-owner-based active probing efficiency.

Hops	IPes	ASes	Organisation
1	169.229.62.1	AS25	
2	169.229.59.225	AS25	Berkeley
3	128.32.255.169	AS25	
4	128.32.0.249	AS25	
5	128.32.0.66	AS11423	Calren
6	209.247.159.109	AS3356	
7	*	AS3356	
8	64.159.1.46	AS3356	Level3
9	209.247.9.170	AS3356	
10	66.185.138.33	AS1668	
11	*	AS1668	GNN
12	66.185.136.17	AS1668	
13	64.236.16.52	AS5662	CNN

Table 2-4 IP-to-AS mappings [53]

[25, 11423]
[25, 11423, 3356]
[25, 11423, 3356, 1668]
[25, 11423, 3356, 1668, 5662]

Table 2-5 vPath of Berkeley University reachability to CNN [53]

2.4.3.2 Experiment

The authors in [20] performed an experiment to evaluate the idea of monitoring reachability among ASes to detect IP prefix hijacking. This experiment consisted of three hosts in three different locations: Seattle, London and Japan. Each host was connected to three BGP routers in order to establish a BGP session with their neighbours and perform hijacking. The host in Seattle was connected to Verio (an ISP) through AS2914, while the host in London was linked to ClaraNet (an ISP) via AS8426; finally, the host in Japan was linked to JPNIC (an ISP) through AS2497. Each of the routers, linked to the hosts, can perform hijacking on other routers. This means that routers represent both attackers and victims at the same time.

Based on the concept of AS-path level analysis, which was generated in section 2.4.3.1 by IP-to-AS mapping that can observe the reachability to other ASes, iSPYes are installed on the three hosts. Afterward, The AS-path levels will construct vPaths like the one presented in Table **2-5** but with the path of the victim. The three hosts graph the prefix-owner's view (vPath) of the victim path before and after hijacking so that the iSPY can detect IP prefix hijacking. This view has to disclose the difference of unreachability among ASes both when and before an IP prefix hijacking occurs. Based on these differences and the size of unreachability to ASes, the iSPY will decide whether or not a hijacking has occurred.

2.4.3.3 iSPY benefits and limitations

Zhang et al. indicate in [20] that the iSPY has many features summarised in the following points: real-time, accurate, lightweight, easily and incrementally deployable, as well as robust in terms of victim notification. Moreover, the iSPY is accurate with a false negative ratio

below 0.45% and a false positive ratio below 0.17%. The iSPY can also probe a large number of ASes and detect hijacking events within a few minutes. Finally, the authors in [20] claim that the iSPY can differentiate between hijacking and link failures based on the size of the cuts (unreachability) caused by the two different events.

The iSPY IP-to-AS mapping approach links IPes to their ASes. However, [54] mentioned that accurate IP-to-AS mapping is a challenging problem due to the lack of a uniform way of numbering router interfaces. In addition, the iSPY uses a BGP routing table to map ASes to their IP prefixes. However, [16] points out many issues involved in getting routing tables from companies, which could make the iSPY undeployable because it cannot be applied. The iSPY could be subject to false positives because it does not have a mechanism for mapping IPes announced by multiple ASes; instead, they will be left unmapped. In addition, iSPY needs to build network reachability to all ASes on the Internet but that is potentially not possible. Comparing to the third proposed method in chapter 5, false negative ratio below 45% and a false positive ratio below 17% are too high as the MOAS conflicts have not been taken into account. The iSPY also does not have a way to differentiate between IP prefix hijackings and link failures. In other words, there is no method to distinguish between unreachability signatures of hijacking and link failures. The authors in [55] point out that it is very difficult to differentiate hijacking form other events based on monitoring the instability of routers. Some middle organisations might prevent access to other ASes by setting up access restrictions on individuals or organisations. In this case, the ICMP will return with several *, but these unreachability marks do not mean the target has been hijacked.

2.4.4 Using origin changes monitoring-based approach

The PHAS (Prefix Hijack Alert System) is an attack detection system proposed to detect IP prefix hijacking in the BGP based on unexpected changes to IP prefix ownership. This system collects its data from Route Views and RIPE and monitors the prefixes of owners when their origin ASes are changed and reports any potential announcement to victims. The PHAS consists of three components: a registration server, origin monitor and local notification filter, as shown in Figure **2.3**. The first component is responsible for receiving registration information (e.g., email addresses and AS numbers, but not prefixes) from users who want to be notified when their IP prefixes have been hijacked. The second component is in charge of monitoring the origin sets for registered prefixes. Origin sets are a combination of ASes that users can use to announce one IP prefix. The monitoring of ownership of an IP prefix is presented in a special notification format; this format will be described in section 2.4.4.2. The notifications are used to send potential hijacking to victims in real time. The users utilise the last component for verifying ownership of a particular origin set (ASes). The notification filter is named local because the filtration is performed by users. This component was added to make the system more user-friendly by observing that not every change in the origin set is necessarily a hijacking [39]. The subsection 2.4.4.1 will discuss the connectivity of these components and explain their functionality in detail.

2.4.4.1 Data collection and detection mechanism

In the first step, users need to register in the PHAS system with unique account. This system is already linked to a server provided with a web-based registration service. This server is allocated for users to register in the system so they can ensure their ownership of the IP

prefixes and be provided with hijacking notifications. Users need to create email addresses and open accounts on the server to communicate with the system in real time. Each user has to have access to the server and update his or her origin set regularly. In addition, the PHAS server supports the PKI service to authenticate and verify users' identities upon sending notifications and checking ownership of an IP prefix. This service is not the core of the mechanism and was discussed thoroughly in different solutions in section 2.3; therefore, it will not be discussed in this section.

In the second step, it was already mentioned in section 2.4.4 that the PHAS is composed of four main components. However, this section discusses the connectivity and functionality of these components as shown in Figure **2.3**. First, the origin monitor needs to be connected to Route Views and RIPE RIS (Routing Information Service) BGP data sources to receive update messages. The origin monitor starts by monitoring changes in the ownership of IP prefixes in the BGP updates. The origin monitor uses a time-window-based mechanism to reduce the repeated reporting of origin change events. Based on the rule of monitoring the changing ownership of a specific IP prefix, the PHAS considers the following conditions anomalous: (1) If an IP prefix appears under another origin AS in an update message, and (2) if an origin AS announces a more specific sub-prefix of another origin AS [39]. Upon finding a potential hijacking, the origin monitor sends, via email, the event in a signed notification message to the user in order to use the local notification filter and confirm that all origin sets (ASes) claiming ownership of an IP prefix are either correct or incorrect. In a circular motion between the PHAS server and the local notification filter, the prefix owner has to ensure ownership of a specific IP prefix before the filter makes the final decision to send a malicious notification. The purpose of using a local notification filter is to reduce the number of false positives sent to the prefix owners because users can have more than one AS but use them to

announce one IP prefix [39]. In other words, the local notification filter checks any change in origin against a locally configured set of valid origins, and only reports an alarm to the user when an unexpected origin change occurs.

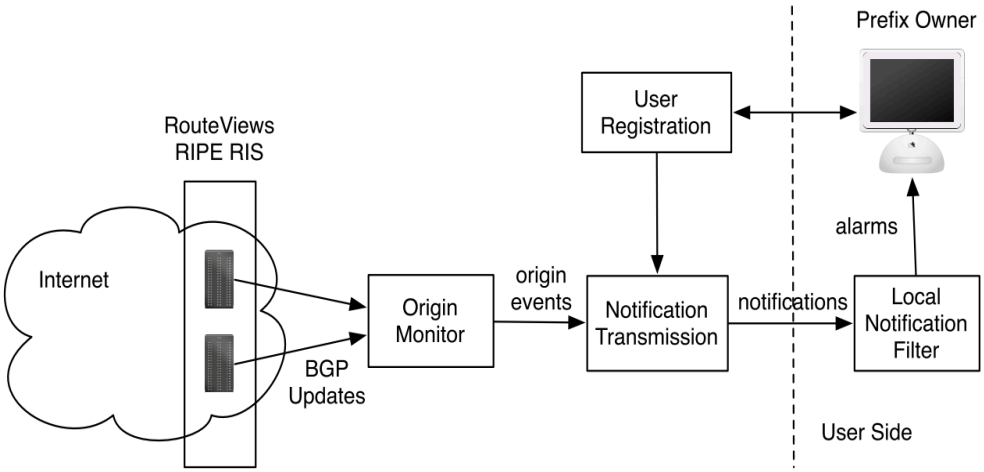


Figure 2.3 PHAS Architecture [39]

2.4.4.2 Experiment

Prefix 60.253.48.0/24 was chosen as an example to monitor its gained and lost ASes in the origin set lists. Parameters ORIGIN-GAINED and ORIGIN-LOST were used to register the changes. The monitoring of 60.253.48.0/24 was conducted between December 21, 2004 and December 28, 2004 [39]. A sample of the notification format was seen by the origin AS monitor as shown in Table **2-6**. The origin AS monitor observed three origin ASes, AS31050, AS29257, and AS23918, claiming ownership of 60.253.48.0/24. This fluctuation indicates the signature of an IP prefix hijacking, which requires the PHAS to send an alarm to the owner of the prefix. However, these claims were considered legitimate changes [39].

<TYPE=gain, GMT-TIME=20041221 04:44:45, PREFIX=60.253.48.0/24, NEW-SET={23918, 31050}, ORIGIN-GAINED=31050>
<TYPE=gain, GMT-TIME=20041221 12:52:33, PREFIX=60.253.48.0/24, NEW-SET={23918, 31050, 29257}, ORIGIN-GAINED=29257>
<TYPE=loss, GMT-TIME=20041221 13:52:49, PREFIX=60.253.48.0/24, NEW-SET={29257, 31050}, ORIGIN-LOST=23918>
<TYPE=loss, GMT-TIME=20041221 13:53:56, PREFIX=60.253.48.0/24, NEW-SET= {29257}, ORIGIN-LOST=31050>

Table 2-6 Notifications observed by the origin AS monitor

Since the prefix owner knew all of these three legitimate origin ASes, the PHAS needed to use the local notification filter to reduce the unfriendly alarms and ignore such notifications using the simple rules shown in Table **2-7**. The prefix owner was responsible for accurate filtration so that the PHAS could precisely detect the IP prefix hijacking.

IF <ORIGIN-GAINED EQ ANY {23918, 31050, 29257} >
THEN REJECT
IF <ORIGIN-LOST EQ ANY {23918, 31050, 29257} >
THEN REJECT

Table 2-7 Rules of filtering legitimate changes

After setting up the rules, the prefix owner would only receive an alarm when the origin changes passed both rules. At 9:30:29 AM on December 24, 2004, such an alarm happened

because the notifications in Table **2-8**, were realised by the filter that AS9121 was not in the list {23918, 31050, 29257} and suddenly showed up. Another alarm was generated to inform the owner that AS9121 stopped announcing the prefix [39].

<TYPE=gain, GMT-TIME=20041224 09:30:29, PREFIX=60.253.48.0/24, NEW-SET={23918 9121}, ORIGIN-GAINED=9121>
<TYPE=loss, GMT-TIME=20041224 11:35:02, PREFIX=60.253.48.0/24, NEW-SET= {23918}, ORIGIN-LOST=9121>

Table 2-8 IP prefix notification alarm

- ***PHAS benefits and limitations***

On the one hand, the PHAS avoids running complex data processing at BGP data collectors; therefore, it can be quickly implemented and run with little overhead at the data collectors. By automating the email processing at the user end, the PHAS provides network operators with real-time alerts in case of the occurrence of potential prefix hijacking. The PHAS is light on the authentication of users because its information is derived from publicly available data; it is also light on data filtering because it does not provide notifications of potential IP prefix hijacking to users after ensuring their ownership of the suspicious IP prefixes. Due to these features, the PHAS is considered very easy to deploy [39].

On the other hand, verifying the correct contact address for each prefix is a challenging problem with no immediately deployable solution [39]. The authors in [39] only mentioned using the PKI but did not describe how it can be linked to the PHAS. As a result, the PHAS

can be considered not to have a mechanism to prevent users from claiming ownership of ASes. In other words, a hijacker can register in the PHAS server and claim that he or she owns a specific origin set. Second, the authors in [38] pointed out that the PHAS does a fine job detecting two different AS numbers claiming one IP. However, it fails to filter out many valid reasons why a network would be a MOAS. Third, a limitation of the PHAS is that it needs to be connected to all BGP data sources on the Internet to protect the IP prefixes of random applicants (users). However, this costs the PHAS server a high computation and a need of infrastructure modification because it does not have a way to predict where the applicants can get the AS number and IP prefix from. Fourth, it is considered centralised detection-based but that will lead to having a difficulty for tracking the changes of IP prefix ownership, in the update messages, in all BGP data sources. Finally, using email to communicate with users in case there is a hijacking is not feasible because the hijacking can happen while the user is away from the Internet or email to confirm the ownership of an origin set. Any detection solution intended to detect IP prefix hijacking has to be decentralised and automated in accordance with the nature of the routers' work.

2.4.5 Using historical-based approach

The PGBGP (Pretty Good BGP) is a developed protocol of the classic BGP proposed to detect IP prefix hijacking in the BGP based on a window of historical routing data. The PGBGP uses two types of information, BGP update messages and the RIB (Routing Information Base), to create the historical window. Routing information from both resources are collected from a developed BGP simulator called the BSIM. The implementation of the PGBGP has two main tasks: first, to construct a set of recently seen data (prefix, origin AS, time) in the historical window; second, to imitate the configuration of BGP operations and

apply a modified decision process of the PGBGP for selecting the best route for each destination prefix [40]. These two tasks will be used to detect prefix and sub-prefix hijacking as described in section 2.4.5.2. Finally, the evaluation of the PGBGP will be discussed in the last section (2.4.5.3).

2.4.5.1 Detection mechanism

The PGBGP uses two sources to detect IP prefix hijacking: BGP update messages and routing tables, which both exist in the RIB and can be generated by the BSIM simulator mentioned in the main section (section 2.4.5). The RIB contains a collection of the best selected routes that can be used by routers to deliver BGP packets to neighbours [22], while BGP update messages contain three important types of information the PGBGP can be based on: the origin AS, which is taken from the last location of the ASPATH attribute; the prefix associated with each update message; and the time each update is received. These two sources are used to constitute a historical window that can show the ownership of every IP prefix. The PGBGP considers the first data received in the window from the two sources as normal. The length of the historical period of the window is represented in days (e.g., 3 days). The historical period should not be very short because that would make many valid origin ASes look suspicious (false positive), or very long because that would allow a repeated prefix hijacking attack to become trusted [40]. The historical window is considered a verification source that can be used in order to check new BGP updates that are announced among neighbours. The historical window is periodically updated with new BGP updates that can pass the check. Any suspicious update message trying to change the state of normal behaviour of the historical window will be quarantined and only accepted after a specific period of time (suspicious period: 1 day) if it is still in the RIB; otherwise, it will be ignored.

If an origin AS and its IP prefix no longer show up in the RIB, they have to be removed from the historical window as well.

New incoming BGP updates are compared to the historical data; if an update message does not pass the examination, it will be considered a hijacking. More specifically, if a new IP prefix in the update message does not have an origin AS existing in the historical window, it will be reported as a hijacked prefix. For detecting sub-prefix hijacking, the PGBGP compares any new prefix in the update message to the recently seen prefixes already saved in the window [40].

2.4.5.2 Experiment

A BGP simulator was developed in [40] to imitate the real functionality of the BGP; this simulator is called the BSIM. The simulator represents the AS topology, route selection and propagation. The PGBGP needs to be integrated with the simulator, which will have 62 core nodes to simulate the action of prefix and sub-prefix hijacking in order to evaluate the accuracy of the detection. The simulator needs to configure five parameters: historical period equal to 3 days, suspicious period equal to 1 day, either a random or core + random deployment type, an attack type which can be a prefix or sub-prefix hijacking, and finally, the number of running the simulation is given 500 times. The first and second parameters are allocated to determine the historical period of collecting data in the historical window, which is explained in section 2.4.5.2, while the third parameter determines the period at which a route is considered suspicious before adding it to the historical window or excluding it. Random deployment is used to enable the PGBGP to work on a random set of nodes, whilst core + random is utilised to enable the PGBGP to function on 62 main nodes plus randomly

chosen nodes. In terms of simulating prefix hijacking, an AS is randomly chosen to originate a prefix and, at the same time, another AS is randomly picked to originate the same prefix. With regard to simulating sub-prefix hijacking, the same scenario for simulating prefix hijacking will be taken, except that the attacking AS will announce a sub-prefix of a victim instead of announcing a super-prefix. Finally, the simulator runs with every simulated attack instance and registers the efficiency of the PGBGP for detecting that attack. With respect to the results, the PGBGP can detect most prefix hijacking attacks but requires large-scale deployment to detect sub-prefix hijacking [40].

2.4.5.3 PGBGP benefits and limitations

The PGBGP is incrementally deployable because it is compatible with the current BGP protocol, requiring changes only to a router's decision rules. Individual ASes have an incentive to adopt the PGBGP, as it provides immediate benefits even when other ASes have not deployed it [40].

In terms of drawbacks, Sriram et al. in [44] claimed that the PGBGP still has some deficiencies because it considers any two different ASes announcing the same IP prefix as malicious; however, this is not true as some organisations have more than one AS. Due to this condition, the number of false positives will be increased in this protocol. In rare circumstances, if an ISP transfers a block of its old provider's address space to a new provider, the PGBGP will treat the routes as suspicious. Determining the AS-level topology of the Internet is a difficult problem [40]. In addition, the PGBGP considers the first collected data in the window as normal in the simulation, but that cannot be applied in reality because it is difficult to predict data that does not include IP prefix hijacking. Using simulation

software (BSIM) instead of the main protocol (BGP) is also not a good choice to detect IP prefix hijacking in the BGP because simulations cannot cover or imitate all routing policy conditions and business relationships, like the ones which exist in real BGP update messages among routers.

2.4.6 Using multiple events monitoring-based

Authors in [38] proposed a system based upon monitoring some factors that could affect the routing of the BGP; based on their effectiveness, the system will try to detect IP prefix hijacking. These factors are summed up as follows: In suspicious MOAS, spam that is sent to attack a specific IP and announcement authorisation. The system chooses a known IP prefix hijacking or used to spam a network and then begins monitoring the change of routing (AS paths). All observations of routing changing that are affected by the factors are correlated to determine if a hijacking is benign or malicious. This section also discusses an experiment applied based on the mechanism; the details of the mechanism will be explained in section 2.4.6.1 and the experiment. The system also bases the detection on different data sources collected by different tools such as traceroute and netflow [56]. Finally, this section ends with displaying the benefits and limitations of the system.

2.4.6.1 Detection mechanism

By looking at the traffic generated by the hijacked prefixes in different sources and using different analytical tools, the authors in [38] expect that their system can detect IP prefix hijacking precisely. The proposed system collects its information from four sources as depicted in Figure **2**.**4**. Based on the changes of network information that spams, suspicious

MOAS, suspicious spamming networks, malicious activities and announcement authorisation make it, the authors in [38] expect that there will be changes in routing (AS paths). First, the system performs monitoring of AS paths (within BGP updates) during an IP prefix hijacking and filters the suspicious MOAS into benign and malicious MOAS output. Similarly, the system receives live spams from spam traps and monitors the changes in routing of the BGP before and after spam that uses a specific hijacked IP prefix. In other words, from the reachability to a particular hijacked and spammed destination, the system monitors the changes of BGP routing using traceroute. The system also uses netflow, which can give a wide array of information about malicious activities that could affect routing, such as large-scale spam campaigns and other scam activities. The authors claim that they expect to observe at least some portions of spam originating from hijacked networks [38]. Finally, the system utilises RIR to notice the announcement authorisations of legitimate ASes that can announce specific IP prefixes. Any one of the five factors can help to detect and distinguish benign from malicious hijacking, but the aim of the system is to correlate all five observations to increase the accuracy of the detection. It is likely that more than one factor can also detect the hijacking of a specific prefix.

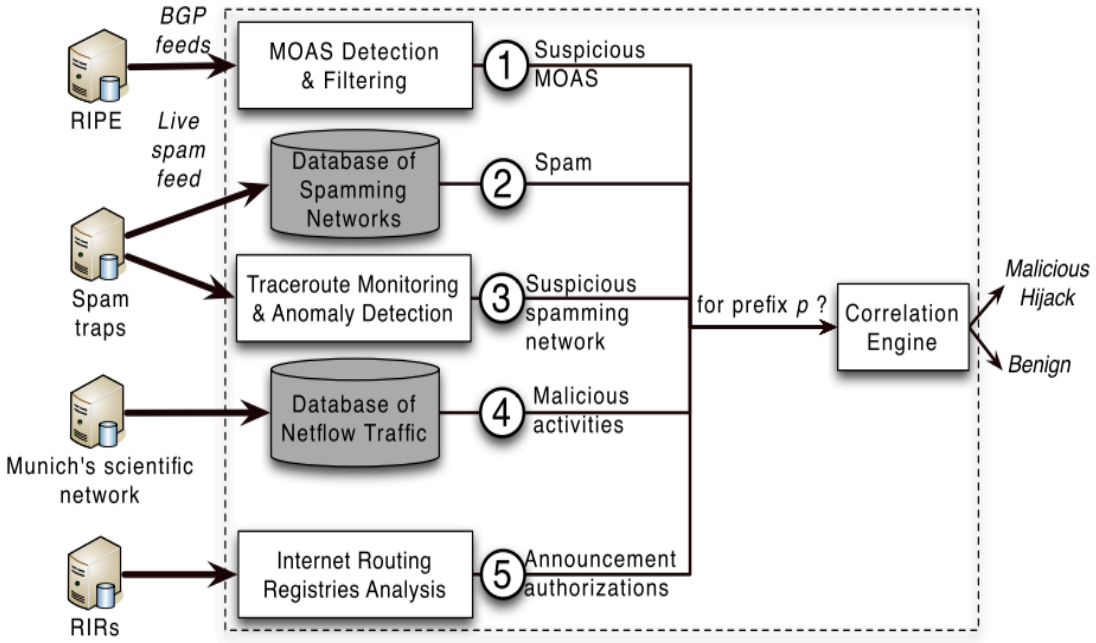


Figure 2.4 System architecture

2.4.6.2 Experiment

The experiment will use the control plane (e.g., in the routing tables and BGP updates) and the data plane to collect desirable data. The control plane provides BGP AS paths, while the data plane can give end-to-end probing information, such as traceroute paths [57]. Routing tables and BGP updates will be collected from a RIPE RIS collector in Amsterdam [58], while traceroute paths will be collected by traceroute. The experiment takes into accounts two factors before detecting real hijacking: normal situations for which multiple ASes can announce the same IP prefix and the analysis time of the traceroute. In other words, before trying to detect a malicious MOAS, the experiment needs to investigate any benign MOAS that can act similarly to a malicious MOAS. Afterwards, the experiment will use active traceroute measurements and start with monitoring BGP AS-level links among ASes in order

to detect real IP prefix hijacking. All normal situations that indicate multiple ASes announcing the same IP prefix need to be removed so they do not turn out to be false positives. Namely, all benign MOAS occurrences need to be removed completely from the monitoring data before using the data to detect a malicious MOAS.

Based on the shared patterns of malicious AS paths, which can be found in BGP updates and in active probing paths (e.g., traceroute paths), the experiment detects real IP prefix hijacking. This method is used especially to reduce the number of false positives that many solutions suffer from during their detection of IP prefix hijacking. With respect to the results, it was observed that about 75% of MOAS events were due to BGP peering relationships, which mean benign MOAS events. The authors in [38] claim that multiple origins of a prefix is not necessarily a sign of a malicious hijacking as they can share a direct AS-level link. The experiment also found that benign MOAS events compared to malicious MOAS events are considered long-lived events. According to what was described in [40], if the threshold of a suspicious MOAS is large enough, the owner of a prefix has enough time to notice the hijacking and take appropriate actions against fake announcements. Instead of using one day for suspicious MOAS events, the authors in [38] used two days as a threshold in order not to limit the time of active probing during the monitoring of suspicious networks. It was also noticed that 14% of non-peering MOAS events lasted more than two days, indicating a malicious MOAS. The authors in [38] argued that it was very difficult to assess the other MOAS conflicts (11%) from the BGP data, which means their nature was unclear.

2.4.6.3 System benefits and limitations

On the one hand, the proposed system can use different sources and data collection techniques and differentiates between suspicious MOAS events. It is also able to observe that 75% of benign MOAS occur among peers and 14% of malicious MOAS appear among non-peers. On the other hand, for 11% of MOAS events, the system can categorise them neither as benign nor malicious. In addition, the authors in [38] indicated that systems based on correlating abnormal BPG events with malicious network traffic are insufficient for conclusively identifying malicious BGP hijacking. They also pointed out that the previously detected cases should again be put to the test, and concluded that state-of-the-art detection systems still have a great chance for improvement in the study of malicious BGP hijacks.

2.5 Comparison between prevention and detection solutions

Regarding advantages, prevention-based solutions, which were discussed in section 2.3, are considered very secure because they require to have pre-knowledge about routing information (e.g. routers and real owners of AS numbers and IP prefixes) are involved in the Internet, while some detection-based solutions do not have that prior knowledge but trace the impersonation of IP prefixes and monitor the behaviour of routers. Both prevention-based solutions can be employed to check the real identities of IP prefix owners and change identification certificates continuously. For example, the soBGP uses a decentralised technique to authenticate AS numbers among routers without the need to trust authorities. Similarly, detection-based solutions have different tools for monitoring sudden, high routing changes among routers using historical windows and predictive statistical methods that can be customised to check the origination of IP prefixes periodically.

With respect to disadvantages, both prevention-based and detection-based solutions have significant limitations. Due to the large-scale exchange of BGP update messages, prevention-based solutions cannot authenticate thousands of messages dynamically and in a short period of time. Because of this limitation, prevention solutions are still considered theoretical proposals and are as such not yet deployable. However, previous detection-based solutions are often based on single or dual combined techniques such as rule-based [16], which requires prior knowledge of network conditions [59] and enormous processing power [47]. Moreover, classification-based, which was discussed in [46], [60], [61], is categorised as a detection-based solution; but in this approach, the accuracy measure of the results is often approximated. In other words, classification-based solutions cannot guarantee whether they cover all possible factors related to a specific issue. Furthermore, a statistical-based approach was also used in [9], [35], [36] which is only suitable for data that are fixed and scaled in time and frequency [47], but these specifications are not applied to BGP data. Finally, detection-based solutions, in most cases, cannot identify the main reason behind the instability of routers because of IP prefix hijacking or other network events such as worms and blackouts [62].

Based on the advantages and disadvantages of prevention-based and detection-based solutions, a dynamic detection method is considered more feasible than a prevention method because it does not require any changes or modifications to the infrastructure of the BGP. As a result, the detection approach will be used in this thesis to detect IP prefix hijacking, as presented in Chapter Chapter : 3, Chapter : 4, Chapter : 5 and Chapter : 6.

2.6 Summary

First, this chapter discussed the security, architecture and communication of the BGP and then discussed the weaknesses that encounter the protocol to give an overview about the BGP. It was found that the BGP has many different categorised vulnerabilities, including being subject to IP prefix hijacking. This event is considered a very serious issue because it enables other ASes to originate the IP prefixes of their neighbours and manipulate the stability of routers, blocking access to services or forcing end users to be redirected to a specific malicious interceptive machine. This chapter also illustrated the IP prefix hijacking scenario, demonstrated the extent of its danger, and discussed its ability to spread widely. Unfortunately, IP prefix hijacking has not yet been solved, as researchers have recently pointed out [63], [64], [65], [4], [38].

In the last two decades, two main security approaches have been proposed to secure the BGP: providing like authentication and verification between peers, and monitoring routing changes that could affect the stability of BGP routers during worm attacks, spamming, and IP prefix hijacking. These two approaches can be crystallised in attack prevention and anomaly detection. On the one hand, previous prevention solutions have had the same objective but have used different mechanisms to achieve trust between routers. For example, the sBGP, soBGP and psBGP authenticate AS numbers and BGP speakers using different types of signing, like centralised or decentralised, and different numbers of certificates, such as one certificate per one AS or one certificate per one BGP speaker. With regard to origin and ASPATH verification, the sBGP and soBGP use multiple centralised levels of signing, while the psBGP does not support trust transitivity. Prevention solutions are also based on trust among ASes or third party organisations, but there is no guarantee that a trustworthy entity

would not manipulate at any time. Furthermore, the nature of the routers' dynamic function and the exchanging of huge numbers of BGP update messages are not compatible with this type of secure BGP.

On the other hand, detection solutions using a rule-based technique to check registered routing information such as unallocated exploited ASNs and IP prefixes for detecting prefix hijacking. This technique can detect hijacking but with many false positives because it does not have a mechanism to differentiate between super- and sub-prefixes of IP addresses within ASes. In addition, monitoring changes in the origination of a specific prefix using different techniques such as historical-based, which was proposed as a detection method but could not detect IP prefix hijacking because it did not take into account that some organisations can have more than one AS number and the first collected historical BGP update messages, that are used to create the historical window, could have prefix hijacking. Moreover, detection solutions observe changes in the stability of routers by monitoring the reachability of specific networks during IP prefix hijacking, but unreachability can occur via different events such as link failures and power outage; therefore, this technique also fails to detect IP prefix hijacking. Finally, detection solutions also monitor the stability of networks during the occurrence of different events, such as spamming, but the authors in [38] state that it is very difficult to detect IP prefix hijacking as correlating abnormal BPG events with malicious network traffic is insufficient to conclusively identify malicious BGP hijacking.

Although statistical-based methods and intelligent research tools such as machine learning and neural network are important, no solution used them to detect IP prefix in particular. For example, the HMM (Hidden Markov Model) is a statistical-based method used in [8] to detect Internet anomalies from BGP data, but does not include IP prefix hijacking. In

addition, Naïve Bayes and SVMs (Support Vector Machines) are machine learning classifiers discussed in [46], [7], [59] to detect worms, but not IP prefix hijacking specifically. Since these tools have not been used, this thesis will build the methodologies for detecting IP prefix hijacking based on them, starting with statistical-based techniques and finishing with the use of machine learning.

Chapter : 3 IP prefix hijacking detection based on statistical analysis

3.1 Introduction

This chapter discusses the first proposed method, which is based on the statistical analysis of variance to detect IP prefix hijacking in the BGP. Determining the resources where update messages can be collected is one of the most important tasks needed to be performed before detecting IP prefix hijacking because some of the resources do not show the content of the BGP packet, like in routing tables, or the route for delivering a specific IP prefix, like in Regional Internet Registries (e.g., the RIPE database). However, update messages can contain both features. Given they contain the values of BGP attributes and policy among routers, they will be used as the resource for raw data. The second task that needs to be explained in this chapter is the study of the data types and the availability of BGP update fields in the BGP packet announcements, as illustrated in section 3.2.2. Update messages have mandatory and optional fields, which might not be sent in some BGP update packets. As a result, each field needs to be investigated in terms of its availability in the packets, the format of its values, and whether the field takes one value or a list of values (different size arrays) in order to find important data involving or which could help detect IP prefix hijacking. The next important task discussed in this chapter is preparing data in an easy analytical format so that different analyses can be applied easily, such as descriptive analyses, machine learning and data mining. This chapter investigates the contents of BGP update messages. The chapter also investigates the relevance of data that could help to detect IP prefix hijacks.

The 2008 YouTube Pakistan IP prefix incident [66], along with other incidents occurring on the same date, will be used as a case study in this chapter. The YouTube incident is

investigated in section 3.2.3 to give an overview of IP prefix hijacking behaviour. The section shows the data state when IP prefix hijacking starts and ends and analyses the hijacking process.

The investigation of BGP update messages, as well as research studies related to the security and vulnerabilities of the BGP, highlight a number of preliminary features to be used for detecting IP prefix hijacking in the BGP. Such features will be based on two main fields, ASPATH and ANNOUNCE, both having a direct relation to the issue of IP prefix hijacking because hijackers manipulate AS origins which exist in the ASPATH field, while the impersonated IP prefixes are available in ANNOUNCE field. Similarly, the number of announcements and withdrawals, which can be extracted from the ANNOUNCE field in BGP updates, can also be used as features to investigate the occurrence of IP prefix hijacking at a specific period of time. Each feature will be tested separately; if there is an indication of IP prefix hijacking, the feature will be considered; otherwise, a new feature will be tried.

In the beginning, it is assumed that IP prefix hijacking can happen in one second but not necessarily spread out fast because BGP routers need time to decide the best path and announce routers to their neighbours; a 15-minute time slot is considered suitable for gathering BGP updates and processing them; more than that would make the processing very difficult, requiring more time because of the size of the BGP update announcements. In addition, less than that is not required because the spreading of fake routes would not be very wide, as will be explained in section 3.2.6. Based on these reasons, 15-minute will be appropriate as a compromise for the size of the sampling data.

There are a number of factors that could affect the stability of edge routers and increase the number of exchanging routing information (BGP update messages) among them. This makes it challenging to distinguish between the stability of routers that could occur due to legitimate events such as blackouts, disconnected big edge routers on the Internet, and any normal configuration operation that could take place by a network operator or malicious events like IP prefix hijacking. However, it is assumed that the behaviour of routers in case of legitimate operations would be different from the behaviour of routers during IP prefix hijacking. Given the limitations of existing methods in identifying an IP prefix hijacking event, this thesis proposes a novel method of detection that is based on observing the behaviour of legitimate BGP updates and updates that could have happened during IP prefix hijacking. Based on the choice of data collection resources and investigations of update messages, a detection method process for extracting relevant features and detecting the IP prefix hijacking issue will be created. Each feature will be independently tested to determine its impact and correlation with IP prefix hijacking. If there is an indication of IP prefix hijacking, the feature will be considered; otherwise, a new feature will be tried. The detection method analyses data and compares the behaviour of routers during incident days and normal days (case study). The results of the analytical process will be discussed in detail in section 3.2.7. The detection method did not work effectively because the assumed benign days of update messages had IP prefix hijacking incidents. In other words, there was no clear scale to know days that were affected with hijacking (abnormal days) from unaffected days (normal days). As a result, the detection method failed to find any indication of IP prefix hijacking. In other words, approximately all days of BGP updates have IP prefix hijacking; therefore, the detection method had difficulty studying the behaviour of routers based on the dates on which IP prefix hijacking occurred.

This chapter is organised as follows: Section 3.2 presents the detection method contents starting from data collection and ending with the implementation of a process for finding an indication of data hijacking to detect IP prefix hijacking. Section 3.3 evaluates the detection method and proposed engine, which is created in section 3.2. The last section summarises the findings of the chapter.

3.2 Detection method

This section discusses a proposed detection method and shows how BGP updates are going to be investigated to detect IP prefix hijacking. The detection method is represented in three main processing components, as shown below in Figure **3**.**1** BGPdump can read MRT format updates and convert them to ASCII. In other words, BGP updates are stored in binary format and a specific organisation; therefore, BGPdump [67] converts the binary format into ASCII. This open source tool is customised to organise all possible data and put them in consistent columns and a unified data-type format (numeric). The second component is called Features Extractors, which extracts useful features from update fields, such as the number of announcements and withdrawals, origin ASes and propagators. The last component is Analytical Processing, which uses descriptive analyses such as calculating mean to put data of normal days against the incident day in a corresponding and consistent way. Analytical Processing also uses standard deviation to work out the deviation between normal and abnormal behaviour of routers. Standard deviation is calculated based on the corresponding routers of the two behaviours on normal days and the incident day. If the deviation is small, this means the similarity is very high between the behaviour of routers on normal days and the incident day; otherwise, it will give a good indication of IP prefix hijacking. The architecture of the detection method is shown in Figure **3**.**1**.

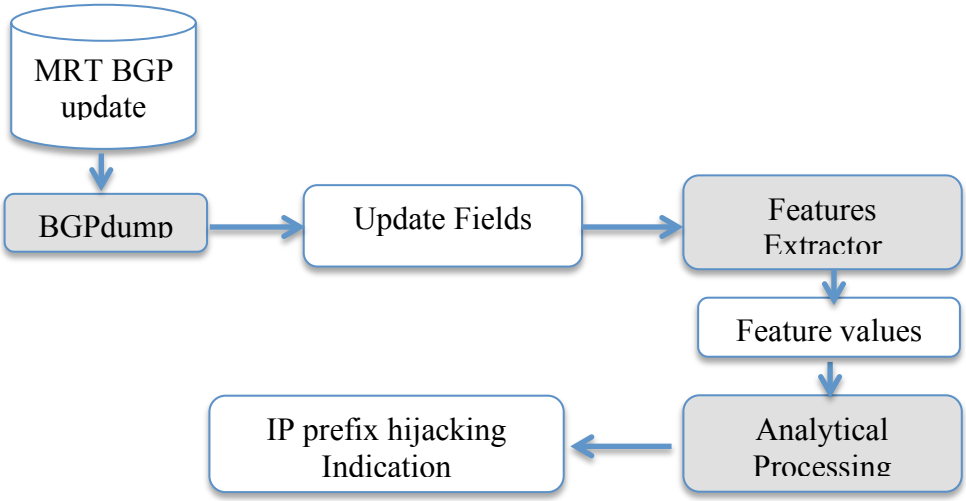


Figure 3.1 Detection method

3.2.1 Data collection

In 1997, Oregon University in the US built a Route Views Archive Project to collect BGP updates intermittently generated by a pool of roughly 50 routers [21]. These updates are stored in MRT format, sorted by dates and UTC times, and listed on the www.Route Views.org website. The updates are downloaded and saved in three separate folders because each day has about 96 BGP update files. The size of update files is not fixed because some routers are added and some of are removed from the Internet. These data are organised to be read automatically by BGPdump and then used later in data analysis.

The reason for selecting the Route Views Archive Project as a data source is that it collects BGP updates in real time and from different locations. These updates are provided for the research community. This source not only registers the routing information of each router separately but also stores BGP attributes and the exchanging routes among routers. BGP

attributes represent the policy of the BGP for choosing the best path to access a particular network; their values are stored in special fields in the updates. The Route Views Archive Project stores data as MRT (Multi-threaded Routing Toolkit) [67] to reduce the size of the update message files. The size of the outputs during the reading of MRT files is reasonable and does not need additional work to split the outputs (ASCII) into smaller files.

BGP updates of an incident (February 24, 2008) one day before (February 23, 2008) and one day after (February 25, 2008) are collected to be used as a case study for detecting IP prefix hijacking in the BGP. This incident is called the 2008 YouTube Pakistan blocking incident, and the two days, one before the incident and one after, are considered the normal days that should not have any IP prefix hijacking incidents. Since the incident lasted only about two hours and no reports talked about any hijacking on February 23, 2008 and February 25, 2008, the two days are assumed hijacking-free. Figure **3**.**2** represents a small snapshot of the format and possible data the case study could have. In other words, this figure shows the format of raw data fields and their data types and structures (e.g. variable or array) of the BGP updates that are going to be used in this thesis.

```
TIME: 05/13/03 00:01:45
TYPE: BGP4MP/MESSAGE/Update
FROM: 134.55.20.229 AS293
TO: 198.32.162.102 AS6447
WITHDRAW
  195.69.188.0/22
  198.153.20.0/22
  203.130.204.0/24

TIME: 05/13/03 00:01:45
TYPE: BGP4MP/MESSAGE/Update
FROM: 134.55.20.229 AS293
TO: 198.32.162.102 AS6447
ORIGIN: IGP
ASPATH: 293 1239 9405 4538
NEXT_HOP: 134.55.20.229
ATOMIC_AGGREGATE
AGGREGATOR: AS4538 202.112.60.250
ANNOUNCE
  219.216.0.0/14
```

Figure 3.2 A snapshot of a raw BGP update message

3.2.2 Data study

BGP updates are either announcement packets or withdrawal packets. Table 3-1 shows all possible data that can appear in different updates. Data from DATE to DESTINATION_AS represent fixed shared fields, which means these fields appear in all announcement and withdrawal packets as exchanging among routers, but they are not BGP attributes. Data from ORIGIN to NEXT_HOP represent mandatory BGP attributes that could show up in every announcement BGP packet. Finally, fields from MULTI_EXIT_DISC to COMMUNITY could show up or be hidden in some BGP update packets when they are announced by edge routers, which makes tracing their values in all packets very difficult. As a result, these fields are not important and ignored because their values are not available in all announcement

update packets. In addition, these mandatory and discretionary attributes do not appear in withdrawal packets; therefore, this type of packet is not considered. The highlighted fields in Table 3-1 are considered the two most important fields relevant to the IP prefix hijacking issue because they include origin ASes and IP prefixes, where the hijacking takes place. Based on these two BGP attributes, features have to be extracted and analysed. Other data can be used as supportive information but are not directly related to the main issue (IP prefix hijacking) such as TIME and DATE to know the exact time and date the IP prefix hijacking occurred, and FROM to know the identity of the hijacker and the propagators to find the affected routers that spread fake routes.

Data in ANNOUNCE	Observations
DATE:	Shared Fields in BGP packets
TIME:	
TYPE:	
FROM:	
SOURCE_AS:	
TO:	
DESTINATION_AS:	
ORIGIN:	Mandatory BGP attributes
AS_PATH:	
NEXT_HOP:	
MULTI_EXIT_DISC:	Discretionary (probably need to take zero)
LOCAL_PREF:	
AGGREGATOR_AS:	
AGGREGATOR_ADDR:	
ORGINAOTR_ID:	
CLUSTER_LIST:	
UNKNOW_ATTR:	
MP_REACH_NLRI:	
MP_UNREACH_NLRI:	
COMMUNITY:	
ANNOUNCE:	Mandatory BGP attributes

Table 3-1 Announcements in BGP update messages

Based on the data analysis and observations of BGP updates, it is noticeable that some fields need to be processed and put into an organised table.

3.2.3 Data preparation

As it was mentioned in section 3.2.1 and 3.2.2 that the most important data to the IP prefix hijacking issue are data related to the two fields, AS_PATH and ANNOUNCE, the research focuses on these two fields, but all possible data in the BGP updates are processed to make them ready upon request. Both data types, mandatory and discretionary in the announcement packets, will be taken, and any attribute that does not have a value will be given zero to keep the consistency of the data among received packets. All non-numeric attributes will be changed to numeric values, and any inconsistent attributes having more than one value will be altered to be consistent, such as ASPATH and ANNOUNCE values. The process of converting non-numeric data to numeric and inconsistent data to consistent is summarised in Table **3-2**. For example, the four-dotted decimal notation of IP addresses is converted to integers using a built-in function in C++ called addr2int to avoid dots in the IP addresses and deal with only numeric data. All these data represented in Table **3-2** will comprise a large table, as a bank of BGP update data, so the detection method can extract relevant features from it.

NO	Data	Numeric and Consistent values
1	TIME	Delete : character to give a numeric value (e.g., 234500)
2	DATE	Delete / character to give a numeric value (e.g., 081913)
3	FROM	Source Router ID is converted to a numeric by removing dots and using addr2int function in C++. AS is removed from source AS to give a numeric value (e.g., AS 6762 = 6762)

4	TO	Destination Router ID = converted to a numeric after taking absolute value Destination AS = 6447 not appear in the ASPATH, 7029 before the last
5	ORIGIN	0 if message type is withdraw IGP=1 EGP=2 Incomplete=3
6	ASPATH	First AS = origin ASes = 6762 ASPATH length = 3 Middles ASes = the number of ASes in between = 1 (in this case) Last AS = last hop/the last AS before the destination (7029)
7	NEXT_HOP	Converted to a numeric after taking absolute value
8	MULTI_EXIT_DISC	Either its original value or zero if it is not sent
9	LOCAL_PREF	Either its original value or zero if it is not sent
10	AGGREGATOR_AS	Either its original value or zero if it is not sent
11	AGGREGATOR_ADDR	Either its original value or zero if it is not sent
12	ORIGINATOR_ID	Either its original value or zero if it is not sent
13	CLUSTER_LIST	Either its original value or zero if it is not sent
14	UNKNOWN_ATTR	Either its original value or zero if it is not sent
15	MP_REACH_NLRI	Either its original value or zero if it is not sent
16	MP_UNREACH_NLRI	Either its original value or zero if it is not sent
17	COMMUNITY	Either its original value or zero if it is not sent Community length = 1
18	ANNOUNCE	Prefixes = 162.39.27.0 Lengths = 24s Packet type = 1 (2 if withdrawn)

Table 3-2 Raw data preparation

3.2.4 2008 YouTube Pakistan incident (case study) analysis

Based on a programme created by the researcher to return with the first packet of an impersonated IP prefix by the hijacker and announced by the owner, and to return with the last packet when the hijacker withdraws the prefix from the Internet, Table **3-3** was generated. This table contains the time of the hijacking, the ASPATH attribute values, and the IP prefix that was hijacked. By tracing where the IP prefix impersonated by the hijacker and announced by the owner over 15-minute time slots, the general behaviour of the IP prefix hijacking can be disclosed. The ASPATH list in Table **3-3** represents one of the BGP

attributes, which contains the ASNs of announcers and propagators to access the destination. Data in the table represent snapshots taken between 12:07:00 UTC and 12:13:07 UTC on February 24–26, 2008, when Pakistan Telecom erroneously announced one of the YouTube IP prefixes. The programme divides the BGP updates into quarters and receives 15-minute time slots per processing. The first 15-minute time slot of the update packet shows the occurrence of the hijack when an edge router belonging to AS17557 announced the 208.65.153.0/24 IP prefix, which was actually owned by AS36561. Since AS36561 could have announced 208.65.153.0/24 at any time, whether before or after the hijacking, it would be helpful to detect whether there are two different routers with different ASNs that announced the same IP prefix, which is an illegitimate action, in the same 15-minute time slot of data. Third, partial data of the update packet in Table **3-3** shows the last period of hijacking activity when the fake route was withdrawn by AS17787.

TIME	Router	ASPATH LIST	ANNOUNCE	WITHDRAW
02/24/08 18:47:57	Hijacker	1280,6461,3491,**17557**	208.65.153.0/24	–
02/24/08 20:51:31	Owner	2497 3549 **36561**	208.65.153.0/24	–
02/24/08 21:01:21	Hijacker	13237 702 17557 **17787**	–	208.65.153.0/24

Table 3-3 Tracking occurrence of YouTube Pakistan hijacking incident

Another shell-scripting programme was written to search for fake routes in all divided update messages (15-minute time slots) on the day of the event, February 24, 2008. The programme already knew the ASNs of YouTube, Pakistan Telecom, and the hijacked IP prefix. The programme divided the BGP updates into quarters and received one 15-minute time slot per processing as in the previous programme, which only looked for starting and ending hijacking. However, this programme tracked all possible hijacking to the YouTube IP prefix.

The programme found that the hijacked IP prefix showed up in five quarters: the 74th, 79th, 81th, 82th and 83th as it is going to be presented and discussed in section 4.2.3.1. The impersonator (AS17557) started by announcing 208.65.153.0/24 (in the 74th quarter) before the legitimate owner. The legitimate AS (36561) began to announce the same IP prefix in the 79th and 81st quarters, but in the absence of the hijacker. The impersonated IP prefix again started to appear in the 82nd and 83rd quarters, but under two different origin ASes, the owner and the impersonator. The hijacked IP prefix was withdrawn by AS17787, which had a direct link to the impersonator. The case study of the YouTube Pakistan Telecom incident is used here to investigate the BGP messaging footprint of an IP prefix hijacking in order to build a reliable, generic IP prefix hijacking detection method. This event will be used as a case study for studying the IP prefix hijacking in this thesis.

3.2.5 Features extraction

Table **3-4** presents a list of features already used in [7], [62], [68], [46], [59] as well as suggested features in this thesis that try to detect IP prefix hijacking in a different way. Preliminary and statistically, some of these features will be used with the detection method to test their indication strength for the IP prefix hijacks. Section 3.2.5 explains how the detection method uses these features while searching for IP prefix hijack indications.

NO	Fields	Features
1	Time	Inter Arrival Time [46]
2		Insertion ASes
3		Deletion ASes
4		Substitution ASes
5		Announcement to Longer Path (max) [7]
6		Announcement to Shorter Path (min) [7]
7		ASPATH_LENGTH per Prefix

8	ASPATH	Average Announcement per Prefix [7]
9		Average AS_PATH Unique Length [46]
10		Average Edit Distance [46]
11		Average of ASPATH_LENGTH per Prefix
12		Average of Number of FIRST_AS per Prefix
14		Average AS_PATH Length [46]
15		Concentration Ratio [7]
16		First Order Concentration Ratio [7]
17		Second Order Concentration Ratio [7]
18		Third Order Concentration Ratio [7]
19		Maximum AS_PATH Length [46]
20		Maximum Edit Distance [46]
21		Number of Repeated ASes
22		Length of Repeated ASes
23	Origin	Number of EGPs [46]
24		Number of IGPs [46]
25		Number of Incompletes [46]
27		Number of Neighbours
28		Number of Specific Routers
30		Number of Updates [7]
31		Summation of AS_SENDER Over Summation of First AS_PATH
32		Summation of NEXT_HOP Over Summation of its Prefix
33	Announcements	Number of Routers Recently Announced [62]
34		Maximum Announcement per Prefix [7]
35		Number of Duplicate Announcements [46]
36		Number of NLRI Prefix Announcements [46]
37		Number of Reachable Prefixes
38		Number of Unreachable Prefixes
39	Withdrawals	Number of Duplicates Withdrawn [46]
40		Number of Implicits Withdrawn [46]
41		Number of Routers Recently Withdrawn [62]
42		Number of NLRI Prefixes Withdrawn [46]

Table 3-4 Suggested and previous solutions' features

According to the conclusion of section 3.2.2, features related to origin, time and withdrawals will not be considered in the thesis. The remaining features will be tested individually, and any one that gives a clear indication between routers' behaviour on normal days and the

incident day will be used in the detection method. For example, insertion ASes measure the number of ASes that have been entered into ASPATH lists in the announcements of hijackers and impersonators. Similarly, depletion and substitution ASes compute the number of ASes that have been deleted or substituted along with announcing routes by a hijacker and victim and returns with their ASNs.

3.2.6 Determining data analysis sampling period

According to the calculation time of spreading fake routes among edge routers, few ASes could be affected within 15 minutes. Any proposed detection method that uses 15-minute as a time slot to search for IP prefix hijacking in the BGP updates would have enough time to notify routers of the hijacking before it spread widely. In fact, the 15-minute processing time can also be a variable to add more flexibility to any IP prefix hijacking detection system. For instance, the time slot can be adjusted to be more or less than 5 minutes. In Table **3-5**, second record represents time, which consists of six digits, first two digits are allocated for hours, second two digits are assigned for minutes and the last two digits are specialised for seconds. If AS18638 and AS3602 are taken as an example to show the maximum time a fake route spreads, the first Q (quarter) shows the first three uniquely affected ASes in the first three records with a maximum of 1 second. Similarly, the second Q shows the first four unique affected ASes in the first six records with a maximum of 3 minutes. The rest of the records (announcements) in the table are considered repetitions of the affected ASes. If the selection of the best path is taken into account, many of the routers in the ASes will not be affected as they might ignore the fake update. In other words, the maximum number of affected ASes cannot be predicted easily. Based on the results of the maximum affected ASes, which is only six ASes, during hijacking incident of AS5571 with AS8190 and AS13902 with AS33694, a

15-minute time slot will be used as a processing period for detecting IP prefix hijacking in the BGP updates. ASPATHs from column 5 upwards represent either Pro (propagator), Ann (announcer) or Pad (padding) for padding zeros which makes columns consistent.

Q	Time	ASes length	Destination ASes	Pro/Ann	Pro /Ann/Pad	Pro /Ann/Pad	Pro /Ann/Pad	Pro /Ann/Pad	Pro /Ann/Pad
1	000700	3	701	25983	18638	0	0	0	0
1	000701	3	701	1239	3602	0	0	0	0
1	000701	7	701	6327	3602	3602	3602	3602	3602
1	000701	3	701	25983	18638	0	0	0	0
1	000703	7	701	6327	3602	3602	3602	3602	3602
1	000712	3	701	25983	18638	0	0	0	0
1	000740	7	701	6327	3602	3602	3602	3602	3602
1	000807	3	701	25983	18638	0	0	0	0
1	000834	7	701	6327	3602	3602	3602	3602	3602
1	000900	3	701	25983	18638	0	0	0	0
1	000928	3	701	1239	3602	0	0	0	0
1	001000	3	701	25983	18638	0	0	0	0
1	001021	7	701	6327	3602	3602	3602	3602	3602
1	001420	3	701	25983	18638	0	0	0	0
1	001447	7	701	6327	3602	3602	3602	3602	3602
1	001513	3	701	25983	18638	0	0	0	0
1	001725	7	701	6327	3602	3602	3602	3602	3602
1	001817	3	701	25983	18638	0	0	0	0
1	001844	7	701	6327	3602	3602	3602	3602	3602
1	001913	3	701	25983	18638	0	0	0	0
1	001940	7	701	6327	3602	3602	3602	3602	3602
1	002007	3	701	25983	18638	0	0	0	0
1	002034	3	701	25983	18638	0	0	0	0
2	002126	3	701	25983	18638	0	0	0	0
2	002200	7	701	6327	3602	3602	3602	3602	3602
2	002220	3	701	25983	18638	0	0	0	0
2	002246	7	701	6327	3602	3602	3602	3602	3602
2	002313	3	701	25983	18638	0	0	0	0
2	002340	3	701	1239	3602	0	0	0	0
2	002406	3	701	25983	18638	0	0	0	0
2	002433	7	701	6327	3602	3602	3602	3602	3602
2	002528	3	701	25983	18638	0	0	0	0
2	002600	7	701	6327	3602	3602	3602	3602	3602
2	002624	3	701	25983	18638	0	0	0	0
2	002651	7	701	6327	3602	3602	3602	3602	3602

2	002718	3	701	25983	18638	0	0	0	0
2	002743	3	701	1239	3602	0	0	0	0
2	002811	3	701	25983	18638	0	0	0	0
2	002837	7	701	6327	3602	3602	3602	3602	3602
2	002931	3	701	25983	18638	0	0	0	0
2	003000	7	701	6327	3602	3602	3602	3602	3602
2	003143	3	701	1239	3602	0	0	0	0
2	003210	3	701	25983	18638	0	0	0	0
2	003236	7	701	6327	3602	3602	3602	3602	3602
2	003331	3	701	25983	18638	0	0	0	0
2	003402	7	701	6327	3602	3602	3602	3602	3602
2	003423	3	701	25983	18638	0	0	0	0
2	003450	3	701	1239	3602	0	0	0	0
2	003517	3	701	25983	18638	0	0	0	0

Table 3-5 Maximum time of spreading fake routes among routers

3.2.7 IP prefix hijacking detection process

After collecting, studying and preparing data, as well as analysing the case study hijacking and finally extracting features, the detection method needs a clear process to detect IP prefix hijacking. The detection method process starts with using the proposed features, which are directly extracted from ASPATH attribute and ANNOUNCE field. These features are organised and sampled in two ways, with the same routers and different routers. The organisation and sampling are shown in Table **3-6** – Table **3-8**. The detection method receives each feature values of the incident day and the normal two days and plots them against each other to display routers' behaviour on normal days and the hijacking day. To find similarities and differences between normal and hijacking days in terms of routers' behaviour, the standard deviation will be calculated.

3.2.7.1 Data organisation of sampling different routers

In the first data organisation of routers' behaviour, the detection method counts different features that represent the behaviour of all routers in different 15-minute time slots of the incident day and IP prefix hijacking-free two days, and each feature is calculated for 50 different routers. For example, vectors in Table **3-6** – Table **3-8** are represented as [F1, R1...R50], [F2, R1...R50], [F..., R1...R50] where F1 is the first feature and F2 is the second feature every 15 minutes, R1 is the first router, and R50 is the last router. Features are tested in the same way: If the results of the features (e.g., the number of announcements and number of withdrawals) work effectively, they are adopted; otherwise, they are ignored. Each value of 15-minute represents the average of different routers' behaviour. Values of routers' behaviour on the incident day are plotted to present routers' stability on February 24, 2008. Similarly, values of routers' behaviour on the normal days are plotted to present routers' stability on February 23, 2008 and February 25, 2008. To compute the similarities and differences between routers' behaviour on the incident day and normal days, the standard deviation of the two behaviours will be worked out in section 3.2.7.3.

Quarters	F1 of R1... R50	F2 of R1... R50	Next feature …
Q1	Mean	Mean	Mean
Q2	Mean	Mean	Mean
Q96	Mean	Mean	Mean
	Mean	Mean	Mean

Table 3-6 Routers' behaviour on February 23, 2008

Quarters	F1 of R1... R50	F2 of R1... R50	Next feature …
Q1	Mean	Mean	Mean
Q2	Mean	Mean	Mean
Q96	Mean	Mean	Mean
	Mean	Mean	Mean

Table 3-7 Routers' behaviour on February 24, 2008

Quarters	F1 of R1... R50	F2 of R1... R50	Next feature …
Q1	Mean	Mean	Mean
Q2	Mean	Mean	Mean
Q96	Mean	Mean	Mean
	Mean	Mean	Mean

Table 3-8 Routers' behaviour on February 25, 2008

3.2.7.2 Data organisation of sampling same routers

The second method for sampling and organising data of routers' behaviour is based on calculating the behaviour of the same routers on one day. On the one hand, the detection method receives one feature for all routers during normal days. For example, router 1 receives a number of announcements per 15-minute and puts them in one column vector. Afterwards, each router computes the mean of the feature (e.g., the number of announcements) to give the behaviour of all routers separately, as shown in Table **3-9** and Table **3-11**, where Q is quarter, F is the feature, and R represents routers. Each value of 15-minute represents the average of the same routers' behaviour. The same process is applied to the hijacking day shown in Table **3-10**. The average behaviour of 50 routers on the hijacking day is plotted against the average behaviour of other 50 routers of the hijacking-free two days. The standard deviation will be calculated for both behaviours, hijacking day and normal days, in section 3.2.7.3 to observe the similarities and differences of routers' behaviour.

Quarters	F1 of R1	F1 of R2	...	F1 of R50
Q1	Number	Number	Number	Number
Q2	Number	Number	Number	Number
Q96	Number	Number	Number	Number
	Mean	Mean	Mean	Mean

Table 3-9 Routers' behaviour for normal day (February 23, 2008)

Quarters	F1 of R1	F1 of R2	...	F1 of R50
Q1	Number	Number	Number	Number
Q2	Number	Number	Number	Number
Q96	Number	Number	Number	Number
	Mean	Mean	Mean	Mean

Table 3-10 Routers' behaviour for hijacking day (February 24, 2008)

Quarters	F1 of R1	F1 of R2	...	F1 of R50
Q1	Number	Number	Number	Number
Q2	Number	Number	Number	Number
Q96	Number	Number	Number	Number
	Mean	Mean	Mean	Mean

Table 3-11 Routers' behaviour for normal day (February 25, 2008)

3.2.7.3 Hijacking and normal routers' behaviour differentiation

This section discusses the way to find similarities and differences in the behaviour of routers in both data organisations presented in section 3.2.7.1 and 3.2.7.2. The detection method plots each data organisation separately and then calculates the standard deviation of both cases. The last rows in all three tables (Table **3-9**-Table **3-11**) represent routers' behaviour. The standard deviation of each router or different routers' behaviour, on normal days and the hijacking day, are taken correspondingly in order to observe the effectiveness of the proposed features. The final result of the standard deviation and the quality of the detection method is discussed in detail in section 3.3.

3.3 Evaluation

The detection method goes through different, important phases to detect IP prefix hijacking, starting by determining the case study and ending by analysing the spread of fake routes. In more detail, the detection method studies data related to the issue to extract the most feasible features and prepares data in a numeric format and a consistent way to come up with an accurate and wider analysis. In addition, the detection method traces the behaviour of IP prefix hijacking and its movement and transfer among announced BGP updates of routers. Moreover, the detection method traces the extent of the spread of fake routes among routers to give an overview of IP prefix hijacking and the appropriate sampling that can be chosen to process update packets. All mentioned data studies, preparations, analyses and traces give the detection method strength to know more about IP prefix hijacking and select the most effective process of organising data to detect IP prefix hijacking.

IP prefix hijacking is a serious issue that could affect the stability of routers continuously, and it is rare to find one day IP prefix hijacking-free. Since it is very difficult to find one day that is IP prefix hijacking-free, the detection method instead needs to separate hijacking packets from unaffected packets, but that requires more effort because each router in the network needs to be checked if it announces other routers' IP prefixes. However, separating hijacking packets from unaffected packets is not a good mean to obtain an identical method because the detection method needs to detect IP prefix hijacking in a normal exchanging order of BGP updates among routers. Two analyses, in section 3.2.4 and 3.2.6, of IP prefix hijacking movement behaviour use the normal exchanging order of BGP updates.

The detection process has the ability to plot the behaviour of routers on normal and hijacking days. However, the detection method process considers normal days (February 23, 2008 and February 25, 2008) IP prefix hijacking-free; but in fact, they are not. Therefore, the detection method could not give a clear view of the difference between routers' behaviour during normal and incident days. In other words, February 23, 2008 and February 25, 2008 were not hijacking-free days, but the detection method considers these two days as hijacking-free because no report was announced or published which pointed out that these days included IP prefix hijacking. As a result, the detection method failed to study the behaviour of hijacking and differentiate it from the normal behaviour of routers. Based on the tested features (the number of announcements and withdrawals), the graphical lines of routers' behaviour on normal days overlap with the graphical lines of routers' behaviour on the incident day, which means no clear indication of hijacking behaviour.

Chapter 4 will present a novel method to distinguish between malicious and benign packets every 15-minute by tracing attack signatures to overcome the limitations of the detection method proposed in this chapter. The new method will search whether there is more than one AS announcing the same IP prefix, instead of comparing the behaviour of routers on incident days with IP prefix hijacking-free days.

3.4 Summary

This chapter proposes a detection method for detecting IP prefix hijacking based on the routers' behaviour during hijacking and normal days for exchanging BGP update messages. The chapter started by picking the case study used with the detection method, then collected data based on that. The case study was the 2008 YouTube Pakistan incident; this incident was

chosen because it is common among reported IP prefix hijacking incidents and covers all possible cases around the hijacking such as both organisations, YouTube and Pakistan Telecom, being in different regions and their routing information being provided by two different RIRs. BGP updates are collected every day by the University of Oregon and put in the Route Views Archive Project for the research community. The detection method downloads updates that include the case study incident.

The second thing the detection method studies are BGP update packets to know which data are critically important to the IP prefix hijacking issue, such as ASPATH, which includes origin and propagating ASes, and ANNOUNCE, which contains IP prefixes. In addition, the detection method prepares and organises data to fit the idea of the detection method. For example, all possible data in the BGP update packets are listed and saved in a consistent way and a unique format to provide organised and tabular data. Based on these data, some features are suggested to explore the behaviour of routers, like the number of announcements, the number of withdrawal insertion ASes, deletion ASes and substitution ASes, as they are displayed in Table **3-4**. The detection method also uses two analyses to study the behaviour of IP prefix hijacking based on the start and end point of the incident and its spread among BGP routers to pick the most appropriate sampling of data.

The detection method has two ways of organising data to detect IP prefix hijacking. This organisation is based on tracking the behaviour of similar routers and the behaviour of different routers separately, as in Table **3-6** - Table **3-11**. The detection method tries features separately. Only two features, the number of announcements and withdrawals, have been attempted, and the detection method directly observed the behaviour of routers on the incident day as very similar and overlapped with the hijacking-free days, which means no

clear indication could show the difference between routers' behaviour during normal times and hijacking time; therefore, the remaining features stopped being tested. The reason behind the similarity and overlapping of routers' behaviour is that February 23, 2008 and February 25, 2008 were considered hijacking-free, but in reality they were not. They had some incidents, which were not reported or published in any papers or sources. As a result, the detection method failed to detect the pattern of IP prefix hijacking.

Chapter : 4 Attack signature and RIR verification-based IP prefix hijacking detection

4.1 Introduction

Chapter 3 proposed a detection method to detect routers' behaviour patterns during hijacking and normal router work. The detection method studies BGP updates and the preparation of data to find the most important data related to the hijacking issue. In addition, Chapter 3 proposed some features and built a programme to determine the effectiveness of these features. However, the detection method failed to test the features and find IP prefix hijacking from routers' behaviour because of the difficulty of finding a case study (one-day BGP updates) hijacking-free. In other words, the detection method failed to find an indication that could distinguish routers' behaviours on normal days (one day before and after the incident day) from the incident day (Pakistan and YouTube). The reason behind the failure is that one day could have many IP prefix hijacks and it is very difficult to find one day free of hijacking. As a result, the detection method cannot detect the IP prefix hijacking based on monitoring the behaviour of routers.

Following Chapter 3, which explores and identifies the limitations of the proposed detection method, a more appropriate methodology to detect IP prefix hijacking will be described in this chapter. The case study of the IP prefix hijacking incident, data study of BGP updates, preparation of the analytical data, the analysis of IP prefix hijacking occurrences, and the selection of sampling data utilised in Chapter 3 will be considered in this chapter with a novel method of organising data and the detection process. This method relies on a signature-based technique and validating results based on the RIR databases. In other words, the detection method searches for more than one edge router, in a specific period of time, claiming the

origin of an IP prefix but not belonging to one AS-block. The detection method uses data reduction and a Binary Search Algorithm (BSA) to detect IP prefix hijacking events quickly. For example, each edge router could announce a specific IP prefix multiple times, but that would lead to the enlargement of the list in which the detection method is searching for IP prefix hijacking; therefore, reducing unwanted and repeated announcement operations is very important to any detection method working with a large amount of data. Based on the BSA, the detection method algorithm traces origin ASes and their actual IP prefixes in 15-minute time slots and categorises the results into two types, benign packets and suspicious packets, which later are validated through RIR databases.

Following the introduction, this chapter presents the structure and architecture of the detection method in section 4.2, which includes four main components. Section 4.2.1 discusses the pre-processing of raw data, while section 4.2.2 talks about extracting organisations' names and their ASes from RIRs, and filtering organisations that announce IP prefixes with one ASN or multiple ASNs to create a verification table. The algorithm of the detection method for detecting IP prefix hijacking is explained in section 4.2.3. Detection method limitations and challenges are also assigned a separate section, in section 4.3, as they are considered very important aspects for evaluating the detection method. The last section is reserved for summarising the chapter.

4.2 Detection method

A new detection method is proposed in this section to identify different ASes that likely announce the same IP prefix and flag these events as IP prefix hijacking. This method is more practical than the previous one proposed in Chapter 3 because of the chance it can check all

BGP-announced packets. It is true that the detection method is self-checking, but because it uses RIRs to validate the outputs, it will lose this feature. In other words, the efficiency of the detection method will depend on the accuracy of the information registered in the RIRs; therefore, it is not easy to predict the results in advance. The detection method consists of four main components: an update processor which extracts ASes and their IP prefixes and organises them, a hijacking detector which will detects IP prefix hijacking, an RIR processor which extracts ASNs and their ORG (organisation) codes for RIRs, and a filter which separates organisations that have one AS from those that have many ASes. The general architecture of the detection method is displayed in Figure **4**.**1**.

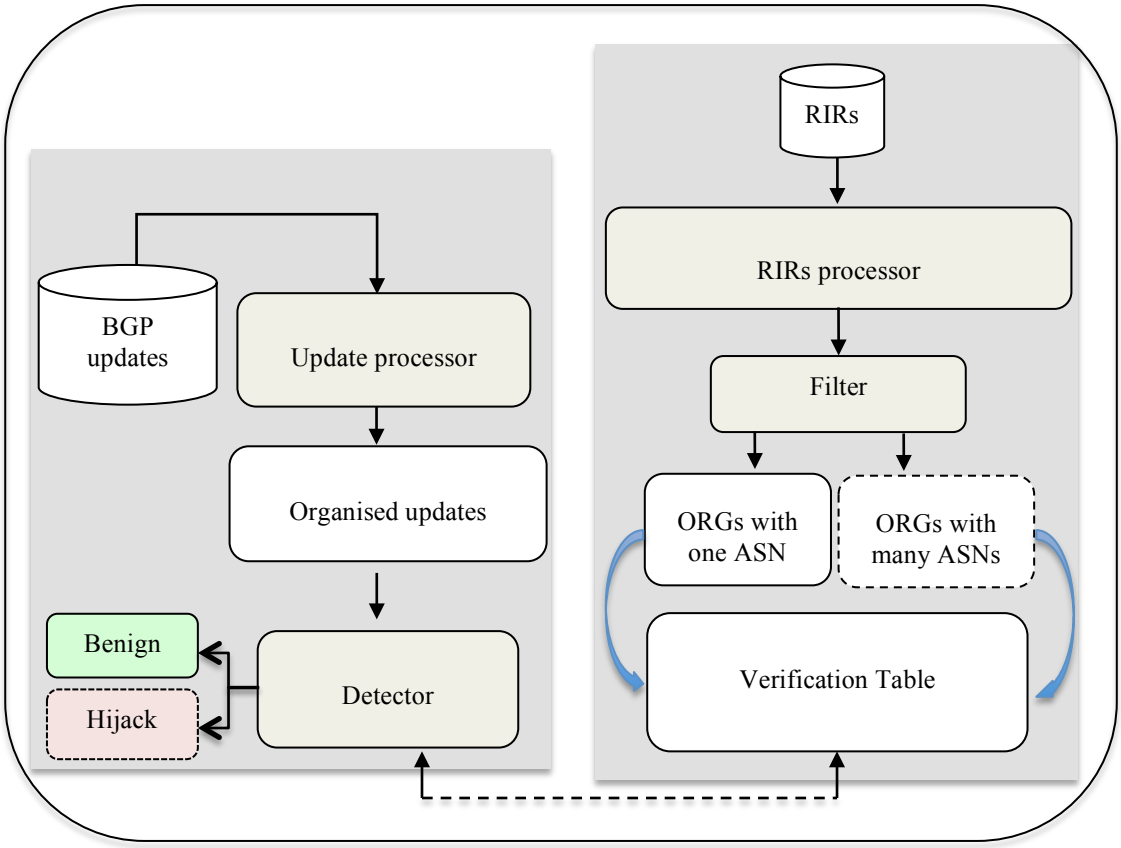


Figure 4.1 IP prefix hijack detection method architecture

RIR processing has an extractor which extracts origin ASes from RIRs and links them to their IP prefixes, and excludes redundant data in order to provide a unified view of repeated prefixes and associated ASes. This component will be demonstrated in detail in section 4.2.2. A filter is used to separate organisations that have one ASes from those that have a block of ASes to make the detection method faster, as will be explained in section 4.2.3. An update processor is linked to BGP updates to extract origin ASes and announced IP prefixes and organise them into associated smaller tables than the one built in section 3.2.3 in Chapter 3. This table will be given to the next component (the detector) to detect IP prefix hijacking. The detector in section 4.2.4 has a single output with two values, either benign or hijacking. Before announcing the last result, the detector will have two types of values in [origin AS, IP

prefix] format and will save them in a SASL (Suspicious Autonomous System List) to be validated through the verification table, which includes organisations with more than one AS number and organisations with one AS number. On the one hand, if an AS exists in ORGs with more than one ASN, the suspicious AS will be removed from the SASL because it is considered a false positive. On the other hand, if an AS exists in the ORGs with one ASN, it will be considered a hijacking. The details of the detection method will be explained in the following subsections.

4.2.1 Update processor

ASPATH and ANNOUNCE data available in the update messages are not consistent in BGP packets, which is likely to make it very complicated to deal with data and trace fake impersonations of routes. Therefore, these data need to be separated into two different organised tables. In other words, ASPATH sometimes appears with different lengths of ASes and ANNOUNCE appears with different lengths of IP prefixes because these two fields are dynamic array data types. In this case, it is very difficult to turn both data types into analytical columns in one table because the columns will be dynamic as well. The potentially relevant data (e.g., ASNs and IP prefixes) to the issue are extracted from the large table, which was already created in section 3.2.3 in Chapter 3, and saved into two different tables as shown in Table **4-1** and Table **4-2**. The first table has ASPATH, including origin ASes, and the second includes IP prefixes. The first table also has ASPATH length, while the second table has the ANNOUNCE length and the prefix range in addition to IP prefixes. Row size of the two tables also has to be equal. In other words, the number of rows in the first table needs to be equal to the number of rows in the second table, and each must correspond because the IP prefixes in Table **4-2** are announced by ASes in Table **4-1**, which will maintain the main

order of the announcement BGP packets. These two tables are created so the detector can associate each origin AS in the ASPATH attribute to its IP prefixes in the ANNOUNCE attribute in the BGP update packets. The update processor in Figure **4**.**1** has the flexibility to change the time slot of processing data; therefore, the detector can work smoothly while checking hijacking in the BGP packets.

Order	ASPATH Length	AS1	AS2	AS ...	Origin	Padding
1	Number	Number	Number	...	Number	Zero
2	Number	Number	Number	...	Number	Zero
3	Number	Number	Number	...	Number	Zero
Last row in time slot	Number	Number	Number	...	Number	Zero

Table 4-1 Organising of consistent, dynamic ASPATH attributes

Order	IP Prefix Length	IP Prefix	IP Prefix ...	Padding
1	Number	Integer IP address	...	Zero
2	Number	Integer IP address	...	Zero
3	Number	Integer IP address	...	Zero
Last row in time slot	Number	Integer IP address	...	Zero

Table 4-2 Organising of consistent, dynamic ANNOUNCE attributes

All of the above tasks and the special organising of BGP updates are considered very important to be performed before detecting IP prefix hijacking; therefore, data pre-processing comes as a first step in the detection method. The update processor treats BGP updates every 15-minute and then saves the results in especial organisation as it is shown in Table **4-1** and Table **4-2**. The ASPATH Length in both tables are needed to access specific AS and IP prefixes during detection processing, while padding is used to maintain the consistency of the ASes and IP prefixes.

4.2.2 RIR processor

This section discusses the process of investigating the organisations that are likely to have more than one ASN and are used to announce their IP prefixes. Since BGP update messages do not have organisation names, the detection method will use RIR databases (specifically the RIPE Whois database [69]) to validate the suspicious ASes that were detected in section 4.2.3. The general structure of processing the verification table is shown in Figure **4.2**. The extractor in the figure searches in the RIR databases for the organisations that have ASNs and collects their codes (unique organisation codes) beside ASNs. If an organisation does not have an ASN in the database, it is simply ignored because it dose not provide the detection method with needed information. The filter in Figure **4.2** separates organisations that have one ASN from those that have more than one ASN because the detection method will validate the results, in the SASL, based on the organisations that have more than one ASN to avoid false positives. ORGs with one ASN need to be filtered out because they do not need to be compared to the data in the SASL, which is going to be implemented in section 4.2.3. The following two subsections, 4.2.2.1 and 4.2.2.2, will discuss these two parts in detail.

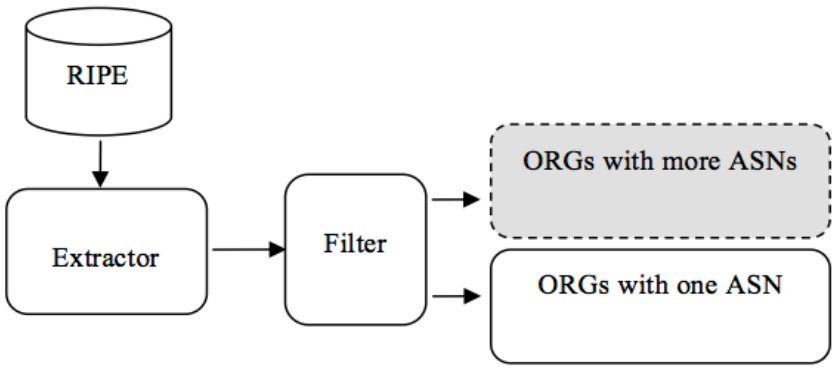


Figure 4.2 Structure of the verification table

4.2.2.1 Extracting organisation codes and their ASNs

As a conditional basis for RIR registration, each organisation has only one code to uniquely identify it. For instance, in RIPE, the ORG-YE1-RIPE code represents Yahoo in Europe, whereas ORG-HBp1-RIPE represents HSBC Bank. In fact, the verification table is processed in three phases: The first phase extracts the ASNs and their organisation codes from the RIPE dump database and stores data in two separate fields, named AUT-NUMS and ORGs. The second phase links AUT-NUMS and ORGs and puts them in several records; for example, AS20535 and its code, ORG-IG12-RIPE, will be in one record (AS20535, ORG-IG12-RIPE), but because the RIPE database sometimes has ASNs without an associated organisation code, the incomplete data will be filtered out. It does inherently limit the capabilities of the presented method because of these missing data. In the last phase, the organisation code is structured as an array created to include all organisation codes that can be mapped to the appropriate ASNs in RIPE.

Organisation codes are divided into three parts (e.g., ORG, IG12 and RIPE, for example) and saved in an array. The second and third indices in the array represent the unique organisation code (e.g., IG12) and data source (e.g., RIPE). Currently, the most important part in the array is the second field because it uniquely identifies the organisations in the same source. The third field of the organisation represents the RIR database name (e.g., RIRs and ASN delegator) that provides the information regarding organisation codes; this helps to differentiate between multiple database source owners in case two have the same middle code (e.g., IG12-RIPE and IG12-ARIN). To optimise the analysis, these two parts are converted to numeric data to suit data in the SASL in section 4.2.3 later.

4.2.2.2 Filtering organisations with one ASN and more than one ASN

Given that the method focuses on organisations with more than one ASN to refine the results, all organisations that have only one ASN will be filtered out because there is no need for them, as will be explained in section 4.2.3. This filtration allows the detector to parse a significantly smaller dataset in order to determine if the suspicious IP prefix events, which were caught in the SASL in section 4.2.3.1, are real hijackings or not. The size of the verification table before the filtering out of organisations with only one ASN was 25 580 records and was reduced to 6283 records through the filter in order to minimise the compared table to make detection faster. If 6283 is divided by 25 580, the speed for detecting the IP prefix hijacking will be improved by 25%. The speed is probably not very clear because the detection method only deals with the RIPE database as an example to test the accuracy and efficiency of the detection method. However, the more databases are processed and used to validate the suspicious ASes, in SASL, the more the speed of the detection is clearer. The detection method will verify its results based on the reduced verification table. The first column in the verification table represents the AS numbers of organisations that have more than one AS, whereas the second column represents the organisation code and data source (e.g., RIPE). The first column will be used as a primary key to be linked to ASNs in the SASL, which will be presented in subsection 4.2.3.1.

4.2.3 Detector

This subsection discusses the last component of the detection method, which has three objectives: First, to map origin ASes to their IP prefixes and save them as pairs in a cell array; second, to remove duplications of associative origin ASes and their IP prefixes; third,

to search for two different ASes announcing the same IP prefix and then, finally, to verify the pairs of one origin AS and IP prefix based on the verification table already created in section 4.2.2 and 4.2.3. The detector receives BGP update messages from the case study (2008 YouTube Pakistan incident, as explained in the introduction of this thesis) every 15-minute and tries to observe the behaviour of the IP prefix hijacking and the other suspicious ASes that could show up.

With every 15-minute BGP update message received by the detector, it associates every origin AS with its announced IP prefixes to generate a cell array composed of pair origin ASes and IP prefixes. Since the origin ASes and IP prefixes in the datasets sampled in section 4.2.1 might be duplicated, the detector first requires a function to remove repetitions of origin ASes and their IP prefixes from the cell array [70] in MatLab. The cell array allows the detection method to map one ASN to multiple IP prefixes. In other words, both columns, origin ASes and IP prefixes in the cell array are subjected to data reduction. The lower the number of ASes and announced IP prefixes, the faster the detection will be. Ultimately, this association, in conjunction with the timing of the message, has to be at the core of an IP hijacking detection method, as it provides all the information about which IP prefixes are apparently owned by their announcing AS routers. The cell array after the reduction of ASes and their IP prefixes and the removal of unwanted data is shown in Table **4-3**.

Order	Unique Origin ASes	Unique IP prefixes
3	137	369760021
4	151	369760021; 369760023; 3697600524
5	174	139438524; 244296124
582	44408	80045022

Table 4-3 Example of the cell array after ASes and their IP reduction

As a second phase and after computing unrepeated origin ASes and their unrepeated IP prefixes, the detector needs to compare the IP prefix of each AS to the IP prefixes of all the origin ASes in the entire time slot (e.g., 15 minutes) to find any IP prefixes advertised by multiple ASes. The detector considers the first row of the origin ASes and its IP prefixes in the cell array as a main row vector. This vector is compared to the further down records in the cell array. The main row vector will be removed from the comparison and the next row vector will be the main row vector to be compared to the following IP prefixes of the origin ASes in the cell array. The processing continues until the end of the cell array. The detector separates each detected event with a row vector [0,0], like in row 3 and 6 in Table 4-4, to differentiate a new suspicious AS. This format of the cell array will be changed in section 4.2.3.1 while displaying the results of suspicious ASes.

Order	Main row vector	Compared row vector	Suspicious ASes	IP prefixes
1	44	4	637	1317679024
2	289	5	5963	1317679024
3	0	0	0	0
4	52	11	747	1536219224
5	285	91	5803	1536219224
6	0	0	0	0

Table 4-4 Example of cell array after comparison

In the comparison phase, the main row vector uses a relatively fast comparison search algorithm called the BSA [71] to compare the IP prefix of the current origin AS to the remaining origin AS IP prefixes. The reason for using the BSA is that it executes array comparisons exponentially faster than other algorithms, such as the Linear Search Algorithm (LSA) [71]. In other words, the BSA is considered the fastest search algorithm because of its speed, which is calculated as O(logn) bits. This specifies the index of an element in the cell

array where n is the size of the cell array. In addition, the detector only applies the BSA to the origin AS that has more than one IP prefix because origin ASes that have a single IP prefix are unique. This reduction of searching for hijacking signatures will also make the detection too fast because unnecessary records to be compared are also ignored. Table 4-4 shows an example of the cell array in two possible formats: unique origin ASes can have either a single IP prefix or multiple IP prefixes. Any suspicious AS caught in the cell array will be saved to the SASL to be validated, and in further sections through the verification table.

4.2.3.1 Suspicious Autonomous System Lists and their analysis

ASLoc and IPPLoc in Table **4-5** – Table **4-9** represent the ASN location and the IP prefix location where the suspicious hijacks were found in the cell array. However, the third column represents the IP prefixes and their range of super- and sub-prefixes. Each row in these tables displays two different origin ASes claiming one IP prefix. From Table **4-5** – Table **4-9**, it is very clear that the detector can detect several different suspicious ASes other than the YouTube and Telecom Pakistan incident. However, this case study incident does not show up among suspicious ASes caught in the cell array, although the hijacking started in Quarter 74, which is represented in Table **4-5**. Namely, the detector skipped the incident because the impersonation of the IP address by two different ASes was not simultaneous in the examined quarter. A proposed method will be discussed in Chapter 6 to address this limitation.

Quarter 74 (starting time of the hijack)					
Order	ASLoc: IPPLoc		Suspicious ASNs		IP prefix hijacked
1	18:3	105:4	637	5963	214.15.201.0/24
2	57:2	424:2	3602	18638	209.5.171.0/24
3	213:2	377:2	9498	17443	202.140.63.0/24
4	446:2	642:2	19750	32004	207.181.144.0/24
5	452:4	492:2	20214	22909	64.139.74.0/24
6	507:7	725:2	23694	38513	202.87.191.0/24

Table 4-5 The suspicious ASes captured by the algorithm in Quarter 74

Table **4-6** also shows that the detector in some quarters can only catch one or probably no suspicious hijackings, as there are no suspicious events flagged out in Quarters 75–78.

Quarter 79 without repetition events					
Order	ASLoc: IPPLoc		Suspicious ASNs		IP prefix hijacked
1	170:2	489:2	10461	35931	65.171.224.0/22

Table 4-6 The suspicious ASes captured by the detector in Quarter 79

Table **4-7** gives another indication that the detector can detect repeated suspicious AS hijackings with different IP prefixes, as shown in Order 2, 3 and 4. The detector can also detect repeated hijackings of the same ASes and IP prefixes, like in Order 3 in Quarter 74 and Order 12 in Quarter 81. In other words, the two same suspicious ASes are found in different quarters.

Quarter 81 without repetition events					
Order	ASLoc: IPPLoc		Suspicious ASNs		IP prefix hijacked
1	73:2	128:2	5571	8190	212.2.0.0/19
2	254:2	498:3	16422	33770	41.223.58.0/24
3	254:3	498:4	16422	33770	41.223.59.0/24
4	254:4	498:5	16422	33770	41.223.57.0/24
5	268:2	498:6	17175	33770	41.220.224.0/24
6	268:3	498:7	17175	33770	41.220.225.0/24
7	268:4	498:8	17175	33770	41.220.226.0/24
8	268:5	498:10	17175	33770	41.220.229.0/24
9	268:6	498:12	17175	33770	196.201.228.0,22
10	328:2	489:2	19750	32004	207.181.144.0,24
11	342:2	421:16	20858	25184	80.75.13.0/24
12	156:3	270:2	9498	17443	202.140.63.0/24
13	351:2	466:3	21396	29606	194.1.150.0,24
14	351:3	466:4	21396	29606	91.199.151.0,24
15	351:4	466:5	21396	29606	195.177.192.0,23
16	383:7	564:2	23694	38513	202.87.191.0/24

Table 4-7 The suspicious ASes captured by the detector in Quarter 81

As can be seen from Table **4-8**, the algorithm of the detector identifies a duplicate announcement of one IP prefix. Both AS17557 and AS36561 announce the same IP prefix, 208.65.153.0/24, of the AS36561 in the Quarter 82. From a detection perspective, this is equivalent to a potential hijacking incident; hence, it was successful in detecting the event.

Quarter 82 without repetition events					
Order	ASLoc: IPPLoc		Suspicious ASNs		IP prefix hijacked
1	142:2	255:186	9229	17557	202.5.150.0/24
2	255:189	500:2	17557	36561	208.65.153.0/24

Table 4-8 The suspicious ASes captured by the detector in Quarter 82

Table **4-9** shows the last period of detecting IP prefix hijacking during the 2008 YouTube Pakistan incident. The hijacking still shows up, as it is presented in Order 3. However, AS17787, which is one of the Pakistan AS-blocks but cannot be displayed in the table because it is not a suspicious AS announcement, withdrew the bogus route from the Internet to return ownership to the real owner (YouTube).

Quarter 83 (last time of the hijack)					
Order	ASLoc: IPPLoc		Suspicious ASNs		IP prefix hijacked
1	339:2	1128:2	10143	38330	203.83.4.0/22
2	447:2	1027:2	13902	33694	208.71.120.0/21
3	549:188	1089:3	17557	36561	208.65.153.0/24
4	705:2	890:2	21792	27169	69.22.144.0/24
5	799:2	822:3	24213	24538	122.200.52.0/24

Table 4-9 The suspicious ASes captured by the detector in Quarter 83

One interesting feature of SASL analysis is that when searching for the 2008 YouTube Pakistan incident, beyond the expected result (the YouTube hijacking), the detector highlighted additional multiple announcement events different from the case study incident. Subsection 4.2.3.2 provides an overview of these identified suspicious incidents in detail.

4.2.3.2 Classifying newly detected suspicious ASes

This section classifies new suspicious IP prefix incidents that were detected alongside the YouTube and Pakistan Telecom incident. Based on the registration information in the RIRs, the SASL is categorised into three categories: the same organisation with multiple ASes announcing the same IP prefix (not a hijacking), different organisations with different ASes announcing the same IP prefix (real hijacking), or no existence to organisation of one or both suspicious ASes (ambiguous event). Examples of these incidents are summarised in the following sections. This section will describe three examples of the categories to determine the most visible method in the detection and validate the results of the SASL while taking into account these categories. The validation will be performed in section 4.2.3.3

Based on a search for the 2008 YouTube Pakistan incident in the BGP update messages, the following list was generated as cases of the same organisation with multiple ASes announcing the same IP prefix:

1 DoD Network Information Centre (DNIC), Comcast Cable Communications Holdings and 24/7 Real Media, in the US [72]
2 MDNX Enterprise Services and MDNX Internet Limited, in the UK [73]

3 Indonesia Network Information Centre, PT Arsen Kusuma and Digital Satellite PT, in Indonesia [73]

Whereas the list below displays some cases of different organisations with different ASes announcing the same IP prefix:

1- Cable Communications Inc. with DH Data Centres Inc. in the US
2- Criteo Corp. with Business Information Group, in the US
3- Townsend Analytics Ltd. with Viztek, Flagler Hospital Inc. with Trident Systems Inc., in the US [72]
4- BHARTI Airtel Ltd. with Karuturi Telecom Pvt Ltd., in the UK
5- NetConnex Broadband with Borwood UK Network, in the UK [73]
6- Pakistan Telecommunication Company Ltd. (Pakistan) with Speed Cast Ltd., in Hong Kong [73]
7- Pakistan Telecom, in Pakistan [72]. and YouTube, in the US [73]
8- Exetel Pty Ltd. and Speedweb Network, Australia [73]

Finally, the cases of no existence to organisation of one or both suspicious AS in the registration information in the RIRs are shown below:

1 New Skies Satellites Inc., in the US [72], [73], with an anonymous AS
2 Afranet Tehran, in Iran [74], with an anonymous AS

In summary, the detection method found some different IP prefix impersonations during the two-hour timeframe of the Pakistan Telecom hijacking. The first three cases in the first example are considered suspicious, and the next eight cases in the second example are

considered real hijackings, while the last two cases in the third example are considered ambiguous because their ASNs do not exist in the IANA, RIRs and delegator ISPs. The number of real incidents, in the range where the 2008 YouTube Pakistan incident shows up, is greater than the number of suspicious and ambiguous incidents (suspicious ASes). After analysing and categorising the pre-detected suspicious ASes, section 4.2.3.3 will discuss the verification of the suspicious ASes of the case study and the new suspicious ASes to show the final results of the detection method for detecting IP prefix hijacking.

4.2.3.3 Verification of suspicious ASes in SASL using the verification table

The verification table implemented in section 4.2.2 will be used in this section with the detector to verify the suspicious ASes caught in the SASL. As mentioned earlier in subsection 4.2.2, the verification table consists of two columns: The first column represents the unique ASNs, and the second column represents the unique organisation codes. Based on the ASNs in the SASL and verification table, the detector searches for two suspicious ASes in the SASL belonging to the same organisation in the verification table. If found, this incident is removed from the SASL because it is not a real hijacking; otherwise, it is considered a hijacking. The detector continues searching until it finishes the candidate ASes. Figure **4.3** shows in detail how, in code, the verification table will be connected to the detector and checks out the results of the SASL to make the last decision.

```
Suspicious = dlmread (suspicious_results);
suspiciouslen = length (suspicious);
VerifDBLen = length(ORGsWithMultiASN);

CASE = 1;
ORGCODE = [1 0; 2 1];

WHILE CASE <= suspiciouslen
    ASN1 = suspicious (CASE, 4);
    ASN2 = suspicious (CASE+1,4);
    CHECK = 1;
    WHILE CHECK < VerifDBLen
          ASN3=ORGsWithMultiASN (CHECK, 1);
          IF (ASN1 == ASN3 OR ASN2 == ASN3)
                IF (ASN1 == ASN3)
                    ORGCODE (1,1)=CASE;
                    ORGCODE (1,2)= . . .
                    ORGsWithMultiASN (CHECK, 2);
                ELSEIF (ASN2 == ASN3)
                    ORGCODE (2,1)= CASE;
                    ORGCODE (2,2)= . . .
                   ORGsWithMultiASN (CHECK, 2);
                END
          END
          IF (ORGCODE (1,2) == ORGCODE (2,2)
                ORGCODE (1,1) == ORGCODE (2,1))
                suspicious(CASE-1: CASE+1,:)=[];
                suspiciouslen= . . .
                length(suspicious);
                ORGCODE (1,2)=0;
                ORGCODE (2,2)=1;
          END
          CHECK= CHECK+1;
      END
CASE= CASE+3;
END
```

Figure 4.3 Verification table linked to the detector

4.2.3.4 Result

In the case of the YouTube hijacking, the detector identified 1767 repeated incidents and 975 unique incidents in the SASL for the whole day. Parsing the analysis through the verification table, the number of suspicious hijackings in the SASL dropped to 969. Table **4-10** shows the excluded suspicious ASes from the SASL from the whole day of the case study incident after using the verification table.

Quarters	ASes	IP prefixes

1	21137 with 25551	145.248.195.0/24
18	21021 with 30824	89.228.16.0/20
18	21399 with 31377	72.246.0.0/22
32	20961 with 21021	87.116.192.0/18
61	8513 with 25228	213.255.206.0/24
87	21021 with 30824	81.190.248.0/21

Table 4-10 Some excluded suspicious incidents from the SASL

4.3 Evaluation

This section discusses three important aspects for evaluating the detector: the limitations, challenges and advantages of the detection method. The difference between challenges and limitations is that detection method limitations discusses the drawbacks of the detection method itself, while challenges refer to difficulties that could be faced by the detection method.

4.3.1 Detection method challenges

The first challenge for the detection method is that it needs to be linked to other databases that have organisation codes to produce a dataset with ASNs, IP prefixes and organisation names, as BGP updates lack them. Adding an organisation name helps to get rid of MOAS conflicts, as one organisation can announce an IP prefix with many ASNs.

Some organisations do not include their unique code (organisation name) in their associated records (e.g., ASN and ORG code) in RIR databases. In other words, every delegator of ASNs and IP prefixes needs to provide their linked organisation names to ASNs in their databases.

From the downloaded RIR dump databases, it can be seen that their formats are not unified. As a result, each RIR database needs to be followed and processed separately to create the verification table. Moreover, delegators of AS numbers and IP prefixes are not few nor fixed, which means new delegators can also provide ASes and IP prefixes; therefore, the verification table needs to be updated regularly.

The huge number of BGP updates being exchanged, the quick, continuous adding and removing of edge routers, and the changing of organisation ASNs and IP prefixes makes detection very complicated for many detection solutions. In other words, the quick, dynamic changing of routing information and registered information in RIRs makes detection very difficult.

Another challenge is that some RIRs do not keep historical records of old Whois registration details so researchers can return to them while collecting completed routing information; therefore, researchers need to build the historical data. Once a record is updated or deleted, the old record is not stored in an archived database.

4.3.2 Detection method limitations

0 The first limitation is that when an edge router impersonates an IP prefix of an AS and the real owner does not announce it in the same 15-minute slot, the detector will not be able to detect the hijacking. In other words, the first 15-minute of BGP update messages received by the detector does not express the first time the owner advertised the IP prefix but a random 15-minute slot after a large volume of routing information has been

exchanged. Thus, it is very difficult to determine if the advertiser of the hijacked prefix is the owner or the hijacker in the absence of another AS in the time slots.

As a first solution, the proposed 15-minute time slot length could be increased to improve detection accuracy, but that would have a negative impact on granularity and potential responsiveness. For example, if the period of the processed data is two hours (the total period of the hijacking but, unfortunately, this period cannot be predicted), the algorithm of the detector could probably detect the IP prefix hijacking from the first compared time slot. However, if the period of the processed data is increased, the short-lived hijacking might take place and finish before the detector detects it because the hijacking will be quicker that the detection. Therefore, It is very difficult to determine the appropriate period of data processing and find the hijacking quickly among the huge number of BGP packets.

Another possible solution is – after comparing ASes and IP prefixes within one quarter – to compare the same quarter to all quarters for one day (cross-validation); but due to the huge amount of data being exchanged, this comparison would affect the performance and speed of the detection method. In addition, the hijackings that could take less than one day will disappear before they are detected.

1 Another significant limitation is that an AS could impersonate a subspace/sub-prefix of a specific IP prefix of another AS, which means two different ASes announce one IP-block; this is also considered a hijacking, but is called a sub-prefix hijacking. For example, YouTube's CIDR (Classless Internet Domain Routing) is 208.65.152.0/22; any impersonator could announce an IP prefix in 208.65.152.0/22-208.65.155.0/32. In other

words, if Pakistan Telecom announces any CIDR between 208.65.152.0/22-32 and 208.65.155.0/22-32, instead of 208.65.153.0/24, it would still impersonate one of YouTube's IP prefixes; however, the detector cannot detect this kind of hijacking without the collaborative method discussed in Chapter 6.

The collaborative method proposed in Chapter 6 is the best solution to avoid these limitations and overcome other possible drawbacks of the detection method. In summary, several copies of the detector have to work with each router in a collaborative way. These copies also need to be connected to routers to receive different time slots of BGP update messages and process them simultaneously.

4.3.3 Detection method advantages

First, the detection method can detect multiple occurrences of the same incident and allows the verification table to identify organisations that announce their routes with more than one ASN. Second, the detection method also achieves objectives regarding detection transparency; the time slot of processing data for detecting hijacking is flexible and gives results quickly. Third, the detection method uses a detection reduction technique to make detection of IP prefixing faster. Fourth, the detection method is considered self-checking as it searches for hijacking signatures in live BGP updates. Finally, the proposed detection method addresses different drawbacks and increases the accuracy of hijacking detection. Generally, the detection method would work to a high degree of efficiency and can detect hijacking when it is provided with live validating data.

4.4 Summary

This chapter proposes an IP prefix hijack detection method using the 2008 YouTube Pakistan incident as a case study to build a trustworthy algorithm to detect hijacking incidents. However, a number of new national or international fake BGP announcements were detected during the aforementioned period, as discussed in section 4.2.3.2. The chapter also discussed a novel approach to distinguish between the same organisations announcing their IP prefixes with multiple ASes and different organisations announcing some prefixes with different ASes in order to avoid false positive detection. This investigation is needed in order to distinguish between normal and malicious BGP operations and address any errors that would likely cause false positives.

From the results in section 4.2.3.4, it is clear that the detection method can work accurately and caught suspicious ASes their organisation codes and ASNs are available in RIPE database. Other RIRs were not included in the verification table because each has its own database structure and would not add many features to achieve the main idea and testing goals of the detection method. If the detection method works with one RIR, then other RIRs will likely work with the method both properly and accurately. First limitation encounters the detection method is determining the period to search for hijacking in the BGP update messages because hijacking behaviour could appear in two different time slots. Second limitation is that if an AS announces an IP prefix in the absence of the real origin AS, the algorithm will not be able to detect the impersonation when it works independently (non-collaboratively). These two limitations will be solved by using the proposed collaborative method explained in detail in Chapter 6. Regarding its advantages, the detection algorithm is

able to improve the accuracy of IP prefix hijacks and remove suspicious hijackings that were already caught and added to the SASL.

Chapter : 5 Detecting IP prefix hijacking based on suspicious Autonomous Systems' connectivity behaviour

5.1 Introduction

An important observation needs to be taken into account regarding the previous solutions when they use ML (Machine Learning) as a detection technique. Previous solutions do not focus on the data, in the raw data, that directly contribute to achieving detection of the hijacking, such as AS numbers and prefixes. Picking the appropriate data to monitor and analyse will help detect IP prefix hijacking. For example, some solutions extract features [46], [7], [62] related to discovering the stability of routers to detect hijacking, but routers can become unstable for many reasons. In other words, any solution for detecting IP prefix hijacking based on monitoring router stability needs to find a way to distinguish between prefix hijacking and other, normal events that cause router instability. For instance, normal connections and disconnections of some routers could cause instability in others. Extracting features based on monitoring the stability of routers cannot determine whether router instability is due to hijacking or normal BGP functionalities. As a result, a detection method will be implemented in this chapter, based on different features, that monitors the connectivity of suspicious routers instead of their stability.

The chapter discusses a novel detection method to detect IP prefix hijacking based on the ML. The detection method implemented in this chapter will use the IP prefix hijack classification to detect patterns of malicious behaviour by tracking the connectivity behaviour of suspicious ASes already found in Chapter 4, section 4.2.3 before verification. The detection method relies on the connectivity behaviour of suspicious ASes towards their neighbours. Namely, the connectivity behaviour of suspicious ASes will be used as input data

to five supervised learning classifiers. In other words, the detection method implements its own connectivity behaviour of suspicious ASes datasets and labels benign and malicious connectivity behaviour based on information about suspicious ASes located in the RIR databases. The detection method features will be extracted from the connectivity behaviour of the suspicious ASes; these features include the number of direct senders and the number of direct receivers for the victim and the hijacker. The quality of features will be calculated before the detection method classifies the suspicious ASes through their behaviour. The results of the five classifiers will be compared to judge the quality of the detection method. If the results of the classifiers are approximately equal, then the classifiers work properly. However, if the results of classifiers are too bad and the quality of features is very good, the limitations of IP prefix hijacking detection will be assigned to the classifiers; otherwise the limitations are attributed to the detection method. The detection method uses a different training dataset percentage and resulted in a 96% accuracy rate for detecting IP prefix hijacking.

This chapter is organised as follows: Section 5.2 presents the components of the IP prefix hijacking detection method. Subsection 5.2.1 extracts some features based on the connectivity behaviour of suspicious ASes. Subsection 5.2.2 talks about a novel algorithm that can compute the similarity of suspicious AS behaviour to explore the capacity for differentiating between benign and malicious router behaviour via the proposed classifiers. Subsection 5.2.3 discusses the methodology of the classification, while subsection 5.2.4 tests and displays the results of the detection method. Section 5.3 evaluates the accuracy of the detection method based on a classification dataset and the outputs of the learning algorithms. Finally, section 5.4 summaries the findings of the chapter.

5.2 Detection method

In the last few decades, Machine Learning has been used to detect anomalies in network traffic and has achieved good results as presented in [75], [48], [7], [46]. Based on these achievements, the proposed detection method in this chapter will use this technique to detect IP prefix hijacking in the BGP. Machine Leaning has different learning approaches for mining data, such as supervised learning, semi-supervised learning, unsupervised learning, reinforcement learning and deep learning. Since supervised learning needs to label malicious and benign instances before performing classification, it is considered more accurate and clear than other learning types; therefore, the datasets of the suspicious AS connectivity behaviour will be structured in a supervised learning format. The IP prefix hijack detection method consists of three main components as shown in Figure **5**.**1**: a Feature Extractor (FE), Labeller and five different ML classifiers.

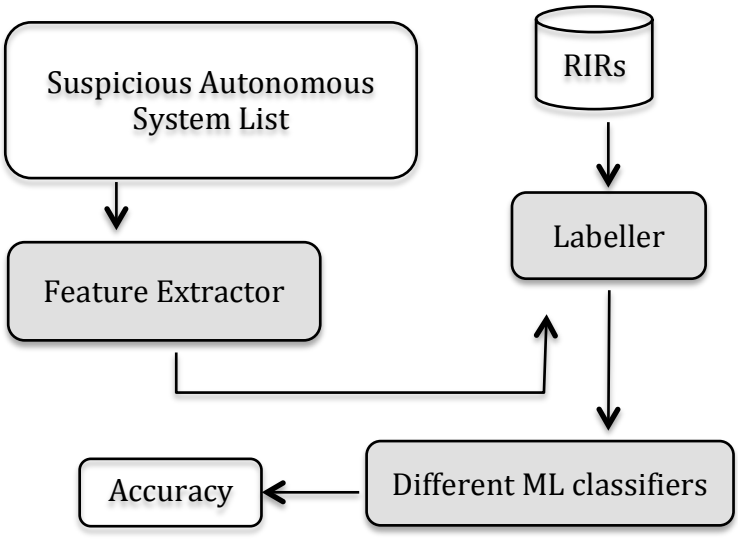


Figure 5.1 Detection method using signature-model-based combination

5.2.1 Features extraction and data sampling

The majority of previous anomaly detection methods extract features based on the stability of routers. However, these methods always fail to detect IP prefix hijacking or are rather poor at differentiating it from other anomalies [7], [8], [46]. Features extracted based on the stability of routers are not practical because routers can become unstable for other reasons, such as blackouts, outages and worms. The detection method proposed in this chapter includes a FE responsible for extracting nine features from potential suspicious ASes connectivity relationships with direct neighbours. All features extracted in this work are novel and extracted in a different way. The extraction method is based on the connective structure (topology) of relationships between suspicious ASes and their direct neighbours. The detection method extracts features from one of the direct locations of the issue, the ASPATH attribute, to differentiate the behaviour of hijackers and victims' edge routers. Table **5-1** displays the proposed features that will be used to detect IP prefix hijacking in the BGP.

NO	Type	Features
1.	Connectivity	Number of repeated incidents
2.		Number of receiver neighbours
3.		Number of sender neighbours
4.		Number of first propagators of suspicious routes
5.		Number of shared receiver neighbours
6.		Number of shared sender neighbours
7.		Number of shared first propagators of suspicious routes
8.		Number of connections between suspicious ASes
9.		Are they neighbours?

Table 5-1 Features of suspicious ASes

Feature 1 in Table **5-1** was extracted based on the observation that unintentional hijacking behaviour, such as misconfiguration, does not impersonate more than one prefix, whereas

man-made prefix hijacks often attack different ASes at the same time. In other words, by monitoring the connectivity behaviour of misconfigured and man-made hijackings, it was found that deliberate hijacks attack different ASes, whereas misconfiguration hijacks attack only one AS. This feature must distinguish between deliberate hijacks and unintentional hijacks. Features 2–7 are based on the connections of the routers to suspicious ASes. Specifically, Features 2–4 focus on the direct neighbours (routers) of suspicious ASes, while Features 5–7 analyse shared direct neighbours between suspicious ASes. Feature 8 and 9 identify direct and indirect connections between suspicious ASes. These features reveal both similar and different patterns of suspicious AS behaviours.

5.2.1.1 Features calculations

In order to see suspicious and affected ASes in the Internet infrastructure and calculate their connectivity when an edge router impersonates the ownership of a prefix owned by another edge router (AS), the NAV (Network Analysis and Visualisation) toolbox [76] will be used in the proposed method. The NAV toolbox can automatically help plotting the topology of suspicious ASes and show their connectivity to their neighbours. Suspicious ASes have a hijacking signature when advertising a BGP update message but are not necessarily announced as a real hijacking, whereas affected ASes spread fake routes over the Internet. The detection method tries to implement the features based on suspicious AS behaviour and not purely malicious or purely benign behaviour because if the detection method can detect IP prefix hijacking from suspicious AS behaviour, it can implicitly classify purely malicious ASes or purely benign ASes as well. Based on the methodology the detection method will use for extracting features, the affected ASes represent the direct neighbours of the suspicious ASes.

Suspicious ASes are composed of two ASes, either benign and benign or benign and malicious. Based on these categories, two types of behaviour need to be worked out for each feature: connectivity behaviour and relational behaviour. Connectivity behaviour is the connection of a suspicious AS to its direct neighbours, whereas relational behaviour is the relation between two suspicious ASes. The relational behaviours of these two suspicious ASes need to be computed separately to differentiate their behaviours.

The connectivity behaviour of suspicious ASes will be calculated based on the suggested features displayed in Figure **5**.**1**, and the results of the two ASes will be subtracted to give the relational behaviour value of the pair of suspicious ASes. For example, we assume that we have a pair of edge routers, (AS1, AS2) and (AS3, AS4), as in Figure **5**.**2**. AS1 has the victim router and AS2 has the hijacker router, while AS3 has the real owner router of an IP prefix and AS4 has another real owner router of the same IP prefix. The victim router and hijacker router in (AS1, AS2) represent real hijacking, while owner routers in (AS3, AS4) are considered suspicious hijacking. The victim receives some announcements from two neighbours, while the hijacker receives its announcements from one neighbour. This shows that upper pair routers in AS1 and AS2 have a different number of sender neighbours, which is 2 and 1 (2 – 1 = 1), respectively. However, the pair suspicious routers in AS3 and AS4 have the same number of sender neighbours, which is 3 and 0 (3 -3 = 0). The owners in AS3 and AS4 are considered suspicious because they carry the signature of the hijacking; two different routers in two different ASes announcing one prefix.

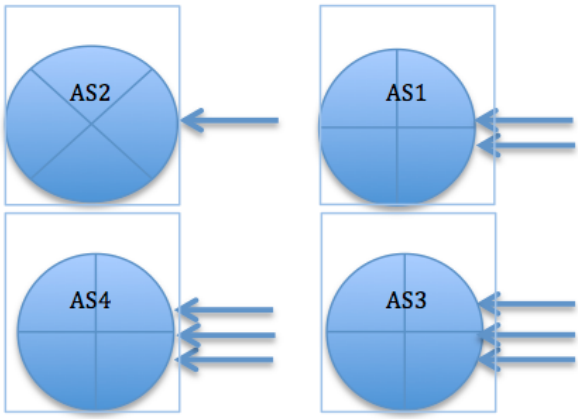


Figure 5.2 Example of one-feature router connectivity calculation

Since the relational behaviour between the two suspicious ASes can be negative in some cases, the absolute value of the differences has to be determined using (**1**. The relational behaviour value is applied to remove the sign. S_{AS1} represents the connectivity behaviour of the first suspicious AS, while S_{AS1} reflects the connectivity behaviour of the second suspicious AS. S_r represents the relational behaviour between two ASes. S_{AS1} and S_{AS2} could both be owners of an IP prefix or a victim and a hijacker.

$$\boldsymbol{S_r = |S_{AS1} - S_{AS2}|} \ (1)$$

The same process for calculating the relational behaviour of a pair of suspicious routers is applied to the remaining features, not just the number of sender neighbours. The results of each pair of ASes are subtracted and put in a column vector to represent the relational behaviour of a pair of suspicious ASes. The column vectors are put together in one training and testing dataset; these column vectors represent the behavioural pattern of the two suspicious ASes.

5.2.1.2 Labelling rules of relational connectivity behaviour of suspicious ASes

After implementing the training and testing dataset in section 5.2.1.1, the relational connectivity behaviour of suspicious ASes will be labelled with malicious and benign marks (0s and 1s) based on RIR database information, such as AfriNIC, APNIC, ARIN, LACNIC and RIPE NCC. Since AS numbers and prefixes are delegated by several organisations and the detection method is only based on the RIRs for differentiating the hijacker from the victim, there are some incidents that cannot be labelled. In addition, some suspicious ASes or impersonated prefixes are no longer available in the RIRs. To label the relational connectivity behaviour of suspicious ASes, the detection method uses three main labels as shown in Table **5-2**.

The strategy for calculating the labels of the relational connectivity behaviour of suspicious ASes in the training and testing dataset is based on four rules: If the RIRs show that both of the suspicious ASes own the route, they are both marked with OWNER and the event is considered benign. However, if the RIRs show that one of the suspicious ASes owns the prefix, it is marked with OWNER while the other is marked as HIJACKER, and the event is considered malicious. If neither of the suspicious ASes owns the prefix, they will be tagged with NOTSURE, and the event will be labelled AMBIGUOUS. Finally, if the RIRs show that the two suspicious ASes do not exist, they will be ignored and not added to the training and testing dataset.

Suspicious AS1	Suspicious AS2	AS1 status	AS2 status	Labels
100	250	OWNER	OWNER	BENIGN
200	10	ATTACKER	OWNER	MALICIOUS
300	50	NOT SURE	NOT SURE	AMBIGUOUS

Table 5-2 Example of suspicious ASes labelling

BENIGN represents semi-hijacks, which means one organisation could own a block of AS numbers and announce one of their prefixes with these ASNs. This announcement gives the same signature of a real hijacking but is in reality benign. MALICIOUS represents real hijacks. AMBIGUOUS events will be removed from the dataset, and only the records labelled BENIGN (represented by 1) and MALICIOUS (represented by 0) will be saved in the dataset, as presented in the next section.

5.2.1.3 Sampling data of suspicious ASes

This section discusses how the relational connectivity behaviour of suspicious ASes and their labelling values are put in supervised classification data structures. Table **5-3** shows a snapshot of the instances calculated based on the proposed features listed in Table **2-2** and the labelling rules discussed in section 5.2.1.2.

Symbols F1–F9 represent the number of features of suspicious AS behaviour, while C indicates whether the event is a hijack or not. Each feature is stored in a separate column vector. These column vectors are concatenated with the class column vector to give a dataset composed of 10 columns, including observation classes, and 340 instances.

F1	F2	F3	F4	F5	F6	F7	F8	F9	C

2	22	1	665	0	1	0	0	0	0
2	0	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	1
1	0	0	66	0	0	0	0	0	0
2	7	0	7	1	0	0	1	1	1
1	5	0	0	0	0	0	0	0	1
10	0	1	0	0	0	0	0	0	1

Table 5-3 Example of supervised sampling data

There are several reasons for making the dataset small and should therefore be discussed in detail. As the registration details of some suspicious ASes are sometimes not shown in RIR Whois during labelling the relational connectivity behaviour of suspicious ASes, it would be very difficult to know hijackers from the victims and label them. Thus, 133 out of 340 records were removed from the dataset. It is likely that the unfound suspicious ASNs or hijacked IP prefixes were delegated to suspicious organisations by other than RIRs, such as big ISPs that have a permission to provide their customers with ASNs and IP prefixes, or these data are no longer used by the organisations. Another reason for not seeing suspicious AS information is because AS-blocks or IP prefix spaces have not yet been allocated by the IANA and attackers might use them for different purposes; but in reality, they do not exist. As a result, the new size of the dataset was dropped to 227 instances. Another important task is to eliminate redundant instances. In other words, all duplicated records were removed from the dataset because there was no need for similar events. In other words, the detection method does not rely on the frequency of an incident's occurrence. After labelling instances based on the RIR Whois validation and removing repeated suspicious observations, the new size of the classification dataset was 113 instances (suspicious AS patterns) by nine attributes (features).

Although the size of the dataset decreased, the hijacking detection performance increased because only the necessary and definite benign and malicious suspicious ASes were

considered, and ambiguous suspicious ASes were removed. By removing duplicated patterns of suspicious ASes behaviour in the dataset, the detection method becomes faster over a large number of BGP updates during live communication between BGP routers and the detection method to detect IP prefix hijacking. It would be a good idea for RIRs or other interested parties (e.g., Oregon University) to store the history of the hijacker and victim ASes in addition to BGP update messages to provide adequate and accurate information to researchers. That would provide interested researchers with a large dataset to work on.

If the proposed learning algorithms in section 5.2.3 can differentiate between patterns of malicious and benign observations, then the extracted features are useful and built in a highly efficient way. Section 5.4 will determine the quality of the features and evaluate the detection method.

5.2.2 Calculating data similarities and differences

A novel algorithm will be implemented in this section to compute the similarity and difference behaviour of benign and malicious route patterns in the classification dataset. The dataset will be sorted according to their classes (0 and 1). Malicious and benign observations will be separated into two different matrices (datasets). Malicious row observations will be correspondingly compared value by value (scalar by scalar) against every benign sequence (instance). Afterwards, the algorithm will create a 0-and-1 matrix for holding the comparison results of corresponding sectors of the malicious row observations and benign row observations. If the corresponding scalars (values) are equal, the algorithm will put 0 in the corresponding location in the new 0-and-1 matrix; otherwise, the algorithm will put 1 in the location. For deciding 0 and 1 values, the algorithm uses the XOR logical operator concept;

the output is true if the inputs are not alike; otherwise, the output is false. One observation of either class (benign or malicious) will be compared to all observations in another class. X_b represents benign matrix row vectors, and Y_m represents malicious matrix row vectors, as in (2) and (3).

$$X_b = [f1, f2, f3, f4, f5, f6, f7, f8, f9] \text{ (2)}$$

$$Y_m = [f1, f2, f3, f4, f5, f6, f7, f8, f9] \text{ (3)}$$

By the end, the algorithm will convert the integer values of the benign and malicious matrices into binary matrices having only two value types, as shown in Table **5-4**. Zeroes represent the similarity scalars, while ones represent the difference scalars between benign and malicious observations in the classification dataset produced in section 5.2.1.3. The first row in Table **5-4** shows the similarity and difference scalars of observations (behaviours) in the benign matrix compared to all observations (behaviours) in the malicious matrix. Each benign row vector creates a 0-and-1 matrix equal to the malicious matrix size [76 X 9], as in Table **5-4**.

	F1	F2	F3	F4	F5	F6	F7	F8	F9
1	0	1	1	1	0	0	0	1	1
2	1	1	1	1	0	0	0	0	1
3	0	1	0	1	0	1	0	0	1
.	.	.	.	.	.	.	.	.	.
76	1	1	1	1	0	0	0	0	1

Table 5-4 Example of 0-and-1 matrix

The algorithm works out the similarities and differences of each matrix separately, as in (4, (5 and (6. First, the algorithm calculates the number of similarities, which is the number of zeros in each row vector, and saves the values in a column vector. Similarly, the algorithm computes the number of ones, which represent the differences, and saves the values in the second column vector against the similarity values. Second, the algorithm calculates the summation of the values in the first column vector, which represent the similarity of the first row vector in the benign matrix to all observations (behaviours) in the malicious matrix. In the same way, the algorithm calculates the summation of the values in the second column vector, which represent the difference of the first row vector in the benign matrix to all observations (behaviours) in the malicious matrix. Equation 4 shows the first part of the computation of similarity and difference of each benign pattern to all malicious patterns, where *for* is the loop starting from the first observation in the benign matrix and ending at the last observation of the matrix. X_{b_i} is benign observations, and Y_m is malicious observations, whereas $\sum 0$ is the summation of similar patterns and $\sum 1$ is the summation of different patterns.

$$_{i=1}^{n} for\{X_{b_i}\ xor\ Y_m\}_{(\sum 1)}^{(\sum 0)} \quad (4)$$

Third, the algorithm takes the means of the summations of similarities and differences, which were already saved in the two column vectors, of benign and malicious patterns, from equation 5 and 6 below, where n is the number of similarities or differences in the column vectors, S_i represents the similarities and D_i the differences of every benign pattern corresponding to all malicious observations in the malicious matrix. $\bar{S}$ returns with the mean of similarity for all benign observations to malicious observations, whereas $\overline{D}$ returns with the mean of differences of all benign observations to malicious observations.

$$\bar{S} = \frac{\sum_{i=1}^{n} S_i}{n} (5)$$

$$\bar{D} = \frac{\sum_{i=1}^{n} D_i}{n} (6)$$

The similarity of malicious and benign patterns was calculated to determine to what extent the behaviour of benign and malicious suspicious ASes are similar. In other words, this calculation shows the quality of the classification dataset (features), which was proposed in section 5.2.1.3. Figure **5.3** shows the similarity of each benign behaviour to all malicious behaviours of suspicious ASes. If the average of benign behaviour against malicious behaviour is computed, it gives 4.31 out of 9, which is the number of features. This average needs to be multiplied by 100 and divided by 9 to give the total similarity between benign behaviour and malicious behaviour of suspicious ASes, which is 47%. This percentage is not very high, which means that the features are adequate to differentiate between the two behaviours.

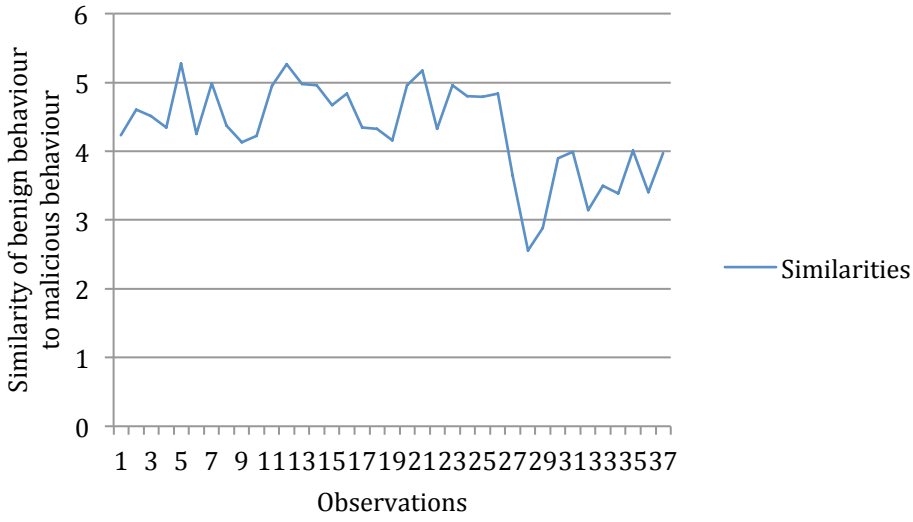


Figure 5.3 Features quality

5.2.3 Classification

This section discusses the process of detecting IP prefix hijacking based on data classification techniques. The specific percentage of the labelled instances (training dataset), as mentioned in section 5.2.1.1, is passed on as inputs to the ML component to be trained; while the other instances, which represent the test dataset (unseen data), remain to be classified. In other words, the detection method uses the Split Test method to divide the dataset already received by the FE and labelled by the Labeller into the training dataset and test dataset. This method helps towards using the training dataset to estimate unseen data and knowing the quality of the proposed features. The detection method tests the quality of the features based on the five classifiers (learning algorithms) will be discussed in section 5.2.3.1. In other words, the detection method uses the best five learning algorithms that have been used on previous applications and which suit the specifications and characteristics of the classification dataset for evaluating the proposed features. The instances (observations or examples) in the classification dataset are classified based on the following steps:

- The dataset is split randomly into 80% training dataset and 20% test dataset for each learning algorithm and retrained several times.
- One of the selected classifiers starts with building the models of different learning algorithms and testing unseen instances of the suspicious ASes.
- Every classifier's parameters are adjusted repeatedly until the best parameters work efficiently with the proposed features.
- The best result of each classifier is registered to be later compared with other good results of the classifiers.

- Based on the offset of the percentage of false positives and false negatives of the classifier results, the best result is announced.

5.2.3.1 Best classifier studies

In 2008, Wu et al. studied the best algorithms that have been used in data mining over the past few decades. The study concerned the best learning algorithms among several methods, such as classification, clustering, statistical learning, association analysis and link mining [77]. Table **5-5** shows the summary of the best 10 supervised and unsupervised learning algorithms that can be used in classification in different applications. However, the detection method is only concerned with supervised learning algorithms because the classification dataset was prepared to work with supervised learning classifiers.

Algorithm	Category	Learning Types	Families
C4.5 (J48)	Classification	Supervised	Decision tree
K-Means	Clustering	Unsupervised	Clusters
SVM	Statistical	Supervised	-
Apriori	Association	Unsupervised	Associations
EM	Clustering	Unsupervised	Clusters
PageRank	Link mining	Unsupervised	-
AdaBoost	Classification	Supervised	Ensemble
KNN	Classification	Supervised	Lazy
Naive Bayes	Classification	Supervised	Bayes
CART	Classification	Supervised	Decision tree

Table 5-5 Top 10 algorithms in data mining [77]

Another study investigating the best learning classifiers was conducted in 2014. The study compared 179 classifiers for 17 families and over 121 different databases and found that the best classifiers are RandomForest versions [78]. RandomForest belongs to a rule-based family and is categorised as supervised learning. In addition, In 2014, Kaur and Chhabra

claimed that an improved J48 was used recently to increase the accuracy rate of classification [79]. J48 is categorised as one of decision tree family that is based on several parameters such as binary split, confidence factor and pruned or unpruned leaves. It can work with datasets that have missing class values, numeric or nominal class and binary class.

5.2.3.2 Classifiers selection

This section discusses the selection of classifiers used with the classification dataset. This selection is based on two previous deep studies of the best algorithms in data mining. The first study [77] was based on the research community and how the best algorithms are used widely in different area in data mining, while the second study [78] was an empirical study performed by some experts in data mining. Both studies are important because they cover each other's limitations. The studies covered two types of learning, but the detection method uses supervised learning algorithms because they can provide a better prior picture of benign and malicious patterns.

The classification dataset was prepared to suit different specifications of the classification algorithms. The dataset is numeric, does not have missing values, has natural and discrete attribute values, fewer data, and is a binary class, which makes the classification of classifiers easier. According to the two strongest attribute evaluators of datasets, PCA (Principal Component Analysis) determined that the dataset has only one redundant attribute and that the remaining eight features are relevant, while the SVM (Support Vector Machine) attribute evaluator considered all attributes to be useful.

Based on the features of learning algorithms, such as accuracy, speed, the offset of having false positive and negatives, and the ability to deal with the structure of the dataset, the classifiers were chosen. The detection method uses J48, which is considered an improved version of C4.5 and C5.0, because it has several advantages and can work with the structure of the dataset. Generally, a decision tree can classify unknown instances quickly and is suitable for interpreting small-sized trees (dataset). In addition, a decision tree can handle discrete attributes, works well in the presence of redundant attributes, and is robust in terms of the effect of outliers; therefore, two classifiers, RandomForest and CART (SimpleCart), were also used. Furthermore, k-NN (k-Nearest Neighbour) and NB (Naïve Bayes) were used because of their ability to classify datasets with only two classes. Both NB and k-NN support complex decision functions or non-linear decision boundaries to isolate multidimensional data and different classes.

5.2.4 Testing and results

Deciding the best classifier for the dataset, as implemented in section 5.2.1.3, was based on the classification accuracy and error rate of false positives and false negatives. This section describes the accuracy classification of the proposed classifiers to classify the benign and malicious behaviour of suspicious ASes in the dataset. The error rate, which shows the best classifier, will be calculated in the two sections below. Since there is no a mechanism in the ML to determine the best percentage, the classifiers were randomly fed with different percentages of training and testing datasets. All classifiers' parameter values in Table **5-6** needed to be changed continuously to suit the dataset characteristics. Based on these changes of algorithm parameter values, the accuracy of the classification was registered and the classification stopped when the classifiers obtained the best results. However, the results in

Table **5-6** do not show the best algorithm that could work with the classification dataset because the error rate was at yet unknown. Thus, the computation of the error rate needed to be worked out for all classifiers, as demonstrated section 5.2.4.2. The highest classification accuracy of the classifiers showed up when the training dataset reached 80%. From the results in Table **5-6**, J48 produced the best result in the classification. However, the detection method took the classifier error rate into account to pick the best algorithm that could work with the classification dataset and the proposed features.

Family	Algorithm	Training dataset	Test dataset	Accuracy
Trees	J48	80%	20%	96%
Lazy	k-NN			91%
Bayes	NB			87%
Trees	CART			95%
Trees	RF			91%

Table 5-6 Results based on Rule and Tree machine learning algorithms

5.2.4.1 Confusion computation

Table **5-7** shows correctly and incorrectly classified instances for both classes, benign and malicious, in the classification dataset. The first row in the confusion matrices column represents the malicious class, while the second row represents the benign class of tested instances (observations) in the testing dataset. None of the algorithms had difficulty classifying malicious observations except Naïve Bayes and RandomForest, but generally all algorithms worked well with the classification dataset and the proposed features.

For malicious observation classification, J48 and CART classified the testing dataset and gave 0 incorrectly classified malicious observations and 18 correctly classified malicious observations, while k-NN yielded 0 incorrectly classified malicious observations and 10

correctly classified observations. However, Naïve Bayes and RandomForest yielded two incorrectly classified malicious observations and 16 correctly classified observations.

In terms of benign observation classification, RandomForest is considered the best classifier among the five algorithms because it had no incorrectly classified observations. J48, NB and CART had the same accuracy rate for detecting benign data, while k-NN was considered the worst as it had two observations classified incorrectly and three classified correctly.

If the total number of chosen classified observations is considered, k-NN only classified 15 observations out of all observations in the classification dataset. On the other hand, CART classified 25 observations, but the remaining three algorithms classified 23 observations. The calculation of false positives and negatives in section 5.2.4.2 will take all of this into account while calculating the trade-off between false positives and negatives of the five classifiers to determine the best one.

Algorithms	Trained dataset percentage	Confusion matrices		False positives and negatives
J48	80%	18	0	0
		1	4	1
k-NN		10	0	0
		2	3	2
NB		16	2	2
		1	4	1
CART		18	0	0
		1	5	1
RF		16	2	2
		0	5	0

Table 5-7 Confusion matrix testing for the five best classifiers

5.2.4.2 False positive and negative computation

From the two equations below, 7 and 8, the detection method can compute the percentage error of the false positives and false negatives for each algorithm, where MF_p represents malicious false positives and BF_n represents benign false negatives. During classification, every classifier randomly selected the number of instances of the testing dataset to be classified. This number represents the total classified observations size.

$$MF_p = \frac{incorectly\ clissified\ instances}{classifed\ \text{observations}\ size} \quad (7)$$

$$BF_n = \frac{incorectly\ clissified\ instances}{classifed\ \text{observations}\ size} \quad (8)$$

Figure **5**.**4** visualises the error rate of false positives and false negatives of each algorithm. This graph shows that false positives (malicious classification errors) are fewer than false negatives (benign classification errors) in the three classifiers (J48, k-NN and CART) and explores the best algorithm based on the trade-offs between false positives and negatives of the classifiers. From the trade-off perspective of false positives and negatives (error rate), the graph shows that J48 and CART are the best two algorithms (classifiers) that can work with the classification dataset and proposed features for detecting IP prefix hijacking because both classifiers have the same value of benign false positives and malicious false negatives.

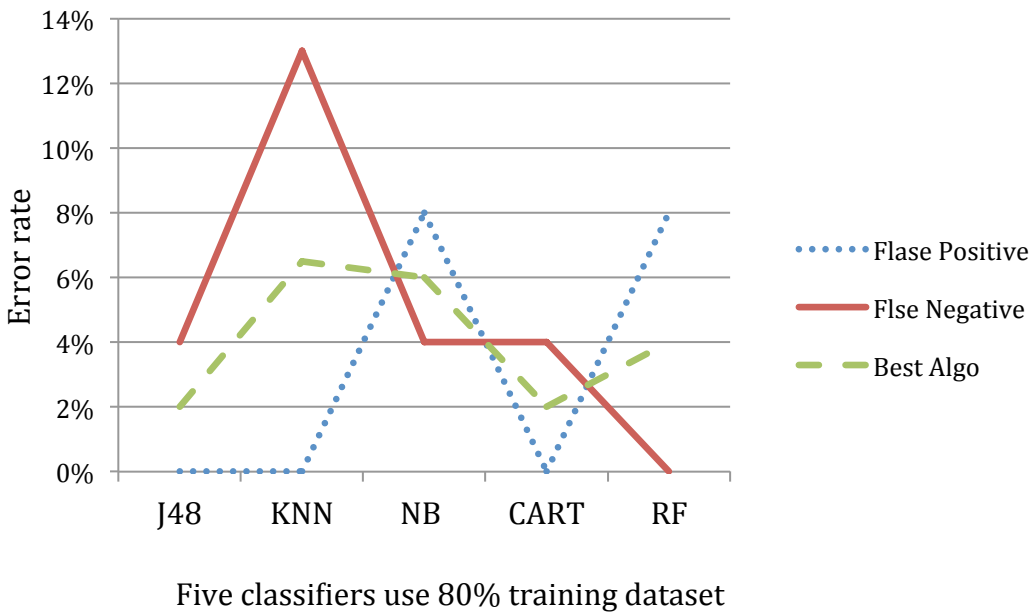


Figure 5.4 Algorithms tried with the detection method

5.3 Evaluation

Comparing to previous solutions such as those proposed in [7], [8], [46], the whole classifiers work in a good efficiency in terms of the detection method because the false positives and negatives do not exceed 13%, as shown in Figure **5**.**4**. However, the two best classifiers that can work with the extracted proposed features are J48 and CART as their error rates are less than the other classifiers. The results of the classification support the 2014 Kaur and Chhabra study, which found that J48 increases the accuracy rate of the classification. However, as J48, k-NN and CART are all good at detecting malicious observations (real hijackings) and RF is considered the best for detecting benign observations (not hijackings), the detection method would work better if the appropriate parameters of RF in the dataset can be combined with the appropriate parameters of J48, k-NN or CART classifiers.

The detection method in Figure **5.5** picks different percentages and tries them on the two best algorithms in order to determine to what extent picking the percentage of the training dataset could affect the result. The graph shows that there is a big fluctuation in terms of choosing the percentage of the training dataset in both algorithms but more so in CART. J48 is more stable than CART over the whole period. As a result, the best choice would be to use 80% of the training dataset to classify benign and malicious router behaviour and then detect IP prefix hijacking.

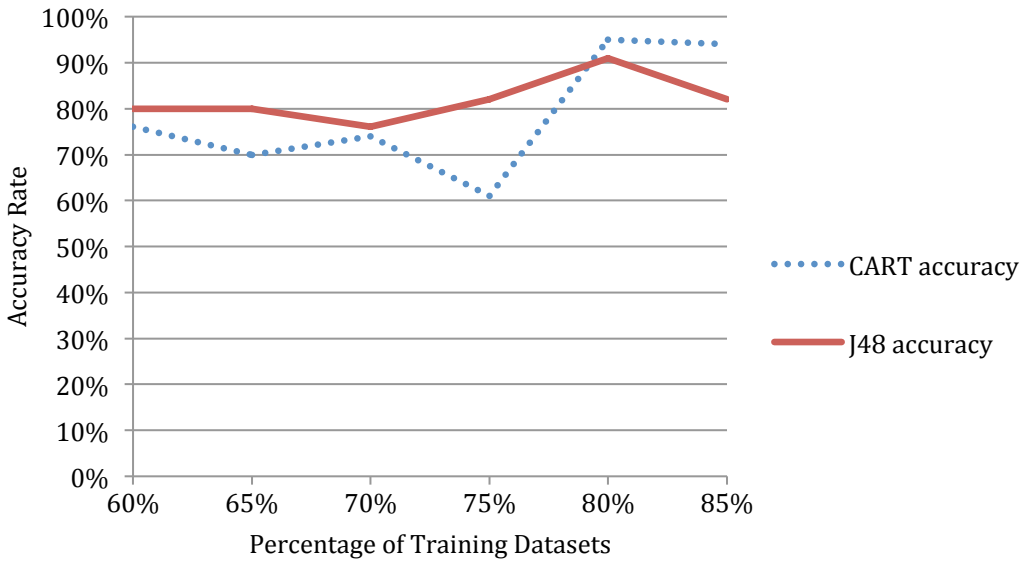


Figure 5.5 Does changing the percentage affect the detection method?

5.4 Summary

In conclusion, this chapter discusses a novel method for detecting IP prefix hijacking in the BGP. The detection method receives suspicious ASes from the signature-based detection method, proposed in Chapter 4, which parses the origin ASes of announcements. The novel method uses machine learning as a tool to classify malicious and benign packets. The

detection method works more deeply and widely because it uses all RIR databases to label benign and malicious AS behaviour. It consists of three components: a Feature Extractor, a Labeller and ML classifiers. The Feature Extractor, or FE, receives suspicious ASes as inputs and extracts features based on the behaviour of suspicious AS connectivity. Feature values of the suspicious ASes are given to the Labeller to be labelled with two classes, benign and malicious. The Labeller uses registered information from the organisations, which is available in RIRs, to identify victims and hijackers and to determine if the suspicious incident is benign or malicious. The outputs of the Labeller, which comprises a classification dataset, are given to different supervised classifiers to predict the behaviour of malicious and benign router behaviour.

There are three main ways to measure the accuracy of the detection method: the total similarity between benign and malicious behaviour patterns (quality of the dataset), extracted features and chosen classifiers. The similarity or differences between the malicious and benign behaviours is calculated in order to examine the quality of the classification dataset before training and classifying them. Based on the similarity or difference between the malicious and benign behaviour patterns, the proposed classification dataset is determined to be passed to the classifiers or not. In other words, studying the total similarity among benign and malicious behaviours in the classification dataset will help to determine the accuracy of the proposed features that attempt to detect IP prefix hijacking. Based on the quality and specifications of the classification dataset were explained in section 5.2.3.2, five learning algorithms are used with the feature values.

The detection method is subject to some conditions that made the size of the classification dataset very small. First, building the dataset and labelling the benign and malicious

behaviour of suspicious ASes were performed manually because the registration information of the suspicious ASes was not available in the RIRs in structured databases to check them in a programmable way. Second, the detector presented in Chapter 4 only depends on the suspicious ASes. Namely, all clearly benign ASes were not included in the dataset because they are considered extra, suspicious ASes contain skeptical benign and malicious ASes as it was explained section 5.2.1.1. In other words, there is no tool to label data from historical incidents accurately. As a result, an accurate dataset was created based on checking the ownership of suspicious ASes and IP prefixes through RIR websites. Generally, the results of the detection method are encouraging compared to previous solutions and because the percentage of false positives and false negatives was less than 5% and the total classification accuracy of the best classifier (J48) reached 96%.

Chapter : 6 Integrating the proposed detection methods with the BGP

6.1 Introduction

BGP is subjected to two types of hijacking: super-prefix and sub-prefix hijacks. BGP routers update their routing tables collaboratively and exchange huge amounts of data in milliseconds. As a result, any proposed detection method based on a detection technique to secure BGP should be structured in a collaborative way so it can search for IP prefix hijacking more efficiently and effectively. This chapter proposes a theoretical framework for collaborative BGP hijack detection. This method is composed of multiple instances of the individual detection nodes, running the method proposed in chapter 4 and 5, to give a chance for cooperation in detecting IP prefix hijacking. In addition, this method aims to overcome two limitations that appeared in the second proposed detection method in chapter 4.

Each detection method instance, in either chapter 4 or 5, can be linked to individual BGP routers to collect data and process them separately. Afterwards, the detection method instances warn connected routers of the occurrence of IP prefix hijacking. In other words, some instances of the detection methods search for IP prefix hijacking similarly and at the same time, but likely with different time slots and BGP updates. A few connections of the same detection method instances to BGP routers in different regions can detect impersonations of origin ASes of other organisations. It is not necessary for all BGP routers on the Internet to be linked to the detection method instances but some of them can detect the IP prefix hijacking.

The aim of the collaboration is to allow the second and third detection method instances to jointly benefit from independently identified events on each router and, subsequently, result in higher accuracy and quicker detection of IP prefix hijacking. An IP prefix hijacking event might not significantly affect traffic exchanged with the impersonated AS until it spreads to multiple/different ASes. To alleviate the effect of the hijacking, the detection method instances of the second detection method have to work collaboratively to prevent the propagation of invalid routes. Similarly, different instances of the third detection method need to work collaboratively to detect hijacking among BGP routers. If there are routers do not have links to the detection method instances, they might be subjected to hijacking. However, their effect will be limited because the other instances of the detection methods, which are linked to the routers, will detect the hijacks and notify other instances with the hijacking. By doing so, each BGP router has a chance to suppress any suspicious routes to prevent itself from further propagating the hijacked routes.

This chapter is organised as follows: Section 6.2 illustrates how the collaborative architecture of the detection method can link BGP routers to collect data and detect IP prefix hijacking quickly and accurately. Section 6.3 discusses a case study of integrating the detection method, proposed in chapter 4, to BGP router and draws a topology of the hijacking and its detection while Section 6.4 discusses the work of the third detection method, proposed in chapter 5, in the collaborative detection method. Section 6.5 evaluates the detection methods, proposed in chapter 4 and 5, when they work collaboratively. Finally, section 6.6 will summarise the whole chapter.

6.2 General architecture of collaborative detection method

This section describes a novel theoretical collaborative detection method that make different instances of the detection methods that proposed in chapter 4 and 5 able to work together and detect different IP prefix hijacking on different routers at the same time. The collaborative detection method consists of separate blocks, with each block having the same instance of the detection method has been built in chapter 4 and 5. The hijacking detection method instances are connected to edge routers, operate independently and categorise network events, but may benefit from sharing detected IP prefix hijackings to detect the effect of such attacks rapidly; therefore, a special remote centralised database (RCDB) is allocated to the collaborative detection method in a trusted centralised organisation (third-party).

The second detection method, proposed in chapter 4, instances need to collect updates and send them to the organisation and the organisation, in return, will have a mechanism to distribute different time-slots of updates (portion of updates) to different instances of the second detection method using unique identifiers.

The third detection method, proposed in chapter 5, instances does not need the organisation to distribute time-slots of updates directly to different instances of the third detection method; instead, the instances will receive the results from the second detection method and send their outputs to the RCDB so other instances of the third detection method can achieve the collaborative detection.

The same detection method instances of either second or third method instances have to work together and their results are controlled by the third-party to improve the reliability and timeliness of the information derived from the BGP update messages. Figure **6**.**1** shows the

general architecture of the collaborative detection method when it is linked to BGP routers and the remote centralised database, and the details of the collaborative detection method will be explained in section 6.3 and its subsections.

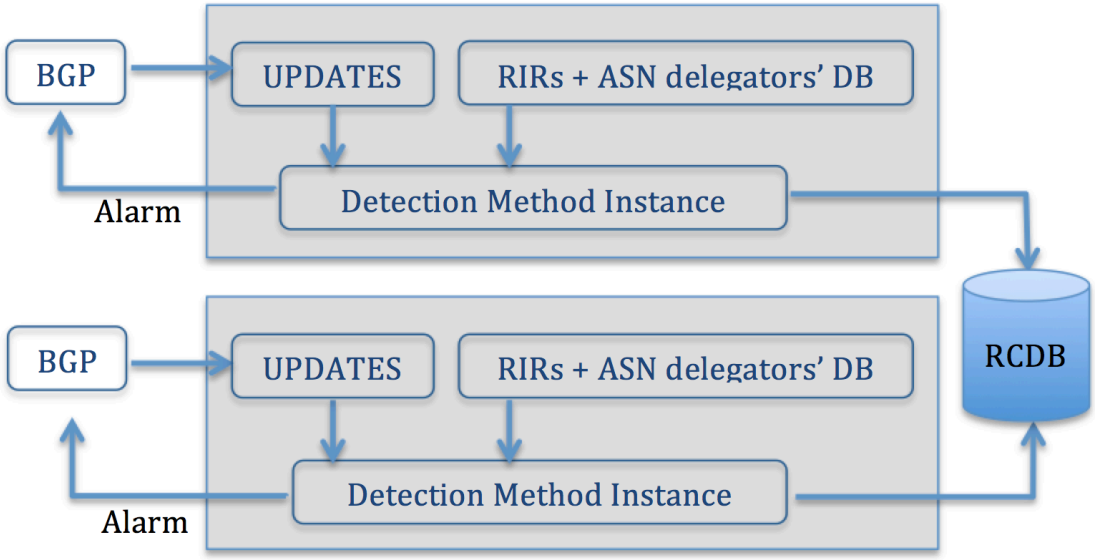


Figure 6.1 Collaborative Detection method architecture

Every detection method instance in the collaborative detection method detects IP prefix hijacking independently and saves the results in the RCDB of the third-party so that other detection method instances can access them for checking the existence of new caught hijackings. The third-party, in turn, originates and updates detected hijacking incidents in order to stop detection method instances notifying old hijacking events to affected network owners (network operators) and to control the size of the database. Afterwards, every detection method should have the ability to check RCDB every one-minute in order to identify whether it has incidents saved by other remote detection method instances. Any detection method instance finds a hijacking incident in the RCDB, whether caught by itself or via other collaborative detection method instances, has to inform it to network operators to be

either removed from their routing tables, to stop propagating it to other neighbours or to ignore the update if it has not arrived yet to unaffected routers.

Both second and third detections will use one topology to evaluate their reliability. The second detection method discusses the use of the proposed techniques that are suggested going to be proposed in section 6.3.2 and the scenario of the hijacking in details. However, the third detection method will only discuss the operations that are different from the second detection method in order to avoid the re-explanation and because the result of the collaborative detection method is same. In other words, the third detection method instances will not talk about using the techniques that already used with the second detection method instances; instead will thoroughly discuss new things.

6.3 Case study of integrating the second detection method to BGP

This section will represent a topology case study to theoretically demonstrate the reliability of the detection method, was proposed in chapter 4, upon IP prefix hijacking is detected. The scenario starts with showing the connections among routers, then injects an IP prefix hijacking and tries to detect it. Afterward, the topology will show how the detection method instances can detect hijacking independently and collaboratively using the RCDB. Previous detection methods suffer from different factors affecting their reliability; therefore, the detection method will evaluate its reliability based on these factors plus other two factors. First, the contents of BGP updates have two attributes (distinguishing AS aggregation and IP confederation) with direct relationship to AS numbers and IP prefixes where the hijacking occurs. Second, MOAS conflicts also considered a problem to the reliability of the previous solutions. Since the proposed detection method deals with the ASes and IP prefixes away

from the distribution policies of ASes and IP prefix management, these attributes and MOAS will not pose any problem to the reliability of the detection method. However, the detection method may encounter two issues regarding detecting the hijacking, the compatibility speed of exchanging routing information and the detection, and missing some incidents when the impersonation of the IP prefix appear in different time-slots (announcements). The topology in section 6.3.1 will show the unaffectedness of the factors, which affected the previous solutions, to the proposed detection method and solve the other two issues by using the collaborative detection method. In addition, connection types (e.g. transit and stub ASes) play an important rule to the reliability of the detection method; therefore, the case study took into account this factor as well.

6.3.1 Topology

The topology of the proposed scenario consists of 5 routers and 5 Autonomous Systems, three network operators, three servers, fifteen connections, one organisation with a Controller and remote centralised database (RCDB), as it is shown in Figure **6**.**2**. According to the ASes contents and their roles, AS600, AS700 and AS800 have three routers but were drawn in this way to show that these ASes are owned by one organisation. In other words, RTD should be allocated to different ASes but it is drawn as one router for the simplicity and to show that one organisation can announce one IP prefix with different ASNs. The organisation owns three ASes (AS600, AS700 and AS800), has one super-prefixes (192.155.10.0/16) and sub-prefixes 192.155.0.0/9-15. RTD in AS600, AS700 and AS800 represents the victim whereas RTB in AS300 represents the hijacker. Three network operators also have been added to the topology in order to show their tasks in the collaborative method. An organisation needs to be assigned as a Controller to manage the RCDB, the distribution of the updates to the detection

method instances and their communication to detect the hijacking efficiently, as it is going to be explained in section 6.3.1.

In terms of the contents of the servers and their roles, Server A has either Cisco, Juniper, or Zebra as a virtual router tool to establish a dummy BGP session (using multi-hop BGP sessions) with RTA in Autonomous System AS100 in order to be able to receive BGP update messages. In like manner, the two servers (server B and server C) should have one of routing tools (Cisco, Juniper, or Zebra) to establish sessions with RTB, in AS300, and RTE, in AS500. All servers should have a copy of the detection method, that already proposed in chapter 4, to show how they are collaborate to detect the hijacking efficiently. The topology also has a RCDB to store IP prefix hijacks that are detected by the detection methods.

In terms of the connectivity among routers, RTA is connected to three nodes, two sending links and one receiving link, which means RTB receives updates from RTA but not vice versa. Similarly, RTC is linked to three routers and receives others' updates through RTD and propagates them to RTA and RTE. However, RTB receives updates from RTA and TRD and sends the updates to Sever C whereas RTD also has three connections but receives updates from tow routers, RTB and RTE, and propagates them to RTC. RTB and RTD can receive and propagate BGP updates from each other. RTE receives BGP updates from RTC and sends them to other two nodes (RTD and Server B). Finally, the three servers are connected to the RCDB in two ways (sending and receiving data).

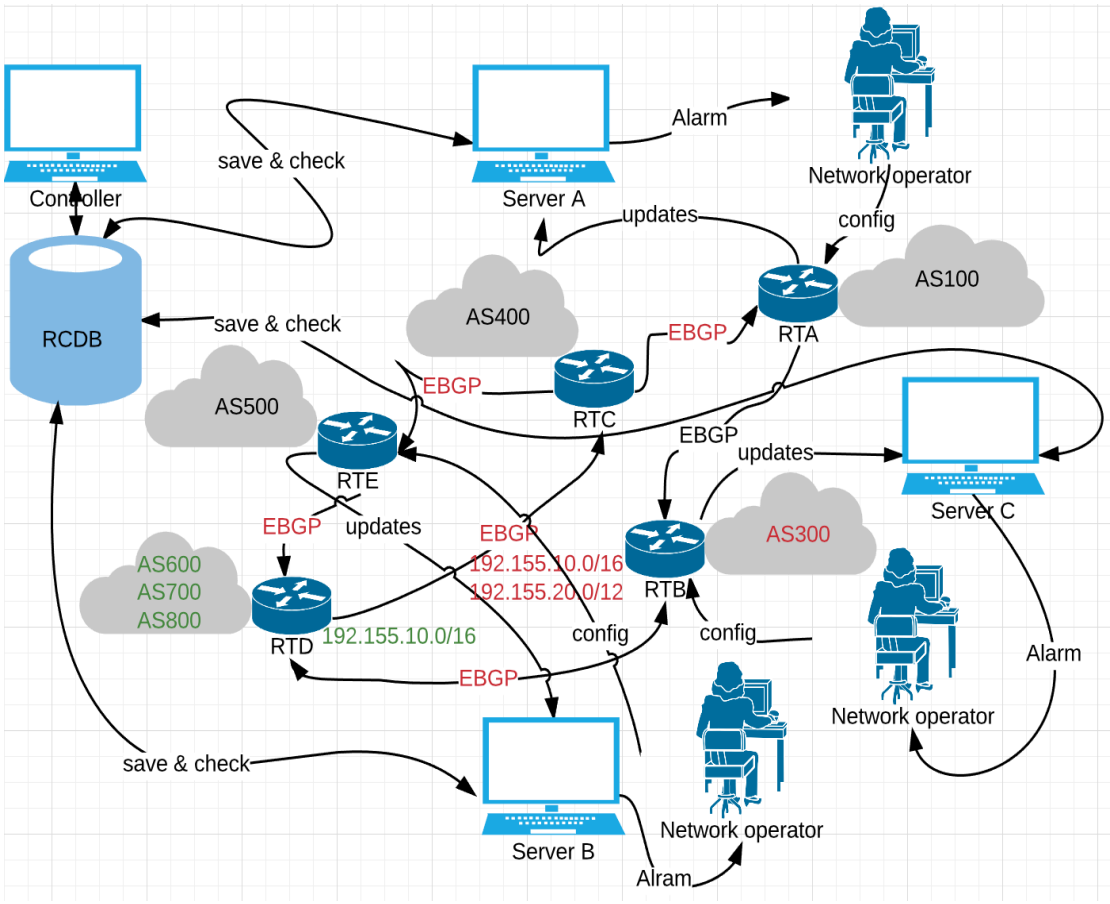


Figure 6.2 Detection method and BGP integration

The RCDB consists of two tables, first table is used for receiving updates from different detection method instances and managing the distribution of the updates to the detection method instances while second table is utilised to manage detected hijackings. The first table has four columns: first column stores the detection method instances' identifiers, second column stores the time of the announcement, third column saves the ASNs while the last column stores the announced IP prefixes. The second table has five columns: first column is allocated for the detection method instances' identifiers but must be unique. The second column represents the times of the hijackings are detected at, while the third column is allocated for the ASes that impersonate other AS IP prefixes. The last column is reserved for the victims' IP prefixes. Fifth column is allocated for notified hijacking. In other words,

when a hijacking is shared with other detection method instances and notified to the network operators, it will be marked as notified so it makes sure it is not going to be notified again by the same detection method.

6.3.2 Mechanism of the Controller and sliding window

The BGP updates are received by different detection method instances, which are linked to the routers. These updates will not be duplicated in the RCDB because each update has different data from other updates. In other words, the RCDB will not have repeated BGP packets but will have repeated announcements (origin ASes, prefixes), which are going to be filtered by the detection method instances, as it was described in section 4.2.3. In addition, the impersonation of the IP prefixes cannot be predicted to which router is going to be happened; therefore it is not feasible to only collect updates from a single route. As a result, the Controller will receive updates from different routers and redistribute different time-slots of updates to the detection method instances. Sliding window technique will be used to solve missing out some incidents that showed up in chapter 4 upon the detection, and centralised controller mechanism will be used to overcome inefficiency between speed of hijacking detection method and routers' speed of exchanging routing information.

The Controller can decide the size of the window based on three things: the specifications of the servers that holding the detection method instances, how quickly the updates are processed by the detection method and the massiveness of saving the exchanging routing information in the RCDB (e.g. 15 time-slots). According to the three criteria, the time-slots might not be equal due to the specification of the servers. First time-slot has to be sent to one of the detection method instances to be processed. The operation is repeated with other

detection method instances continuously but with excluding one BGP packet from the previous time-slot to achieve sliding window technique. For example, if the time-slot is 5 minutes, the first detection method will be given 5 minutes time-slot of updates and the second detection method will be given a time-slot based on the specification of the server the detection method linked to, but starting from the second update. In this case, the collaborative detection method can search for IP prefix hijacking signatures on different detection method instances without missing any incidents or having an issue to the speed of hijacking detection. Any detection method detects a hijack has to save it in the RCDB in order to the detection method instances be able to notify other routers with the hijackings.

In the context of the RCDB maintenance, the Controller needs to manage the RCDB in order to prevent repetition of announcing the same hijacking by the same detection method instance upon checking the database for new hijacking and prevent the growth of the database. The Controller updates each detection method records in the RCDB as soon as it detects a new hijacking and the hijackings became out of date. If more than one detection method instance has the same IP prefix-hijacking incident in the RCDB, the incident will be given a high threshold to decide whether the incident is a real hijacking. In other words, the reliability of the detection can be achieved based on the number of detection method instances that announcing the same hijacking.

6.3.3 IP prefix hijacking and detection method instances collaboration

The topology drawn in section 6.3.1 has three servers and these servers have virtual routing tools (e.g. Zebra) that can work as BGP protocol and receive different updates from the routers. Based on the Controller policy and sliding window technique, proposed in section

6.3.2, the Controller redistributes updates to the detection method instances on the servers in order to allow them searching in specific data in parallel for increasing the speed of the hijacking detection and to prevent omitting any incidents during detection processing.

The RTD router in the topology represents the victim router; the router starts with announcing 192.155.10.0/16 using different Autonomous Systems (AS600, AS700 and AS800). Since AS600, AS700 and AS800 are considered transit ASes to AS400 and AS300, TRC and RTB can receive the announcements and propagate them to their direct neighbours. Second detection method instances are previously installed on Server A, B and C and based on the Detector, which implemented in chapter 4, the instances can realise that 192.155.10.0/16 is announced by different ASes, which indicates to the hijacking signature. However, since the detection method uses the Verification Table proposed in chapter 4, the detection method instances notice that AS600, AS700 and AS800 belong to one organisation. As a result, the detection method instances will not save the event, in the RCDB, as a hijacking but rather will ignore it. Based on the Detector algorithm, the detection method instances will remove all repeated announcements before detection processing.

RTB represents the hijacker, which announces the super prefix 192.155.10.0/16 or the sub-prefix 192.155.20.0/12 of RTD. RTB will announce the prefixes to Server C and to the victim itself while RTD will spread the hijacking to RTC, then to RTA and then to RTE. However, RTD will not be able to detect the hijacking because it lacks the security and it is not linked to any servers has a detection method instance. In other words, since RTD does not have the detection method, it will not be able to detect the hijacking but other routers, which are linked to Server A, B and C, can stop receiving the fake route and drop it, if it is already saved in the routing table by network operators. Likewise, the detection method might not be able to

detect the hijacking from the first time because RTD could announce its IP prefixes (192.155.10.0/16 or any sub-prefixes) at a different time-slot while the detection method searching for hijacking. However, since the collaborative detection method is going to use different sliding windows, on different servers, the hijacking will be detected as soon as the real owner announce the prefixes and then will be saved in the RCDB.

The topology shows the importance of the collaborative detection method, as the routers are directly connected and receiving updates from RTA, RTB, and TRE will not be affected continuously and spread fake routes like RTC and RTD. For example, RTA will stop spreading the hijacked prefix to the hijacker itself or any router beyond the typology could receive the update from it because it already has been notified with the fake route. In addition, RTE will be affected by the hijacking through RTC but because it is connected to the RCDB and have the detection method installed on Server B, it can either stop receiving the malicious announcement or remove it from its routing table and will not spread it to the victim again. However, the victim (RTD) will keep propagating the malicious announcement because it receives it from the hijacker (RTB) continuously. The RTD will not realise the hijacking although one of its prefixes was impersonated. In same manner, RTC will keep receiving the malicious announcement because it is linked to an affected router that does not have the detection method. In this case, the hijacking will be spread between RTD and RTC or to routers that might be connected to them behind the topology. RTC will keep spreading the hijacking to RTA and RTE till the hijacker stop impersonating the IP prefixes of RTD but both routers (RTA and RTE) will ignore the malicious update because it has been detected that the prefixes are belonging to RTD not RTB. Based on the time and the identifiers in the second table in the RCDB, the detection method instances on Server A, B and C need to checking the RCDB every one-minute in order to find out if there is new hijacked prefixes to

be ignored, removed from their routing table or stop propagating them. Finally, the detection method instances can either stop the hijacking independently by finding the hijacking separately or collaboratively through finding the hijacking in the RCDB.

6.3.4 Notifications with hijacking

The collaborative detection method can help the detection method instances to detect hijackings and notify them to network operators, but it does not take the action of removing the hijackings or even stop them. In other word, this is not a limitation of the design, but a limitation of scope because the detection method instances are not intended to stop the hijacking, but only to identify the hijacks. When operators are provided with the hijackings, they need to ignore, remove fake routes from their routing tables, or stop propagating them to their neighbours. The network operators can do so either manually, by injecting withdrawal hijackings or automatically, by making a separate programme to receive fake announcements are going to be sent from the detection method instances.

6.4 Case study of integrating the third detection method to BGP routers

The third detection method, which was proposed in chapter 5, uses the same facilities of the topology that is in section 6.3.1. However, the second detection method, propped in chapter 4, traces the signature of the hijacking while the third detection method, proposed in chapter5, tries to find the behaviour patterns of the benign and malicious routers. Therefore, the third detection method will use the models behaviour of the routers that were given by the five classifiers to notify the network operators. First, Server A, B and C will have instances of the third detection method along to instances of the second detection method, which proposed

in chapter 4. The second detection method instances receive BGP updates using one of routing tools such as e.g. Zebra and give the SASLs results to the third detection method instances. The third detection method instances need to compute the behaviour patterns of suspicious ASes in the SASLs based on the features that proposed in section 5.2.1.

The behaviour patterns need to be changed every ten minutes as the third detection method is based on the second detection method and needs to allocate enough time to data processing and IP prefix detection. Each second detection method instance needs roughly five minutes to collect updates and two minutes to give the SASL while a third detection method instance needs around three minutes to give the behaviour patterns of the benign and malicious routers and then calculate the malicious ASes. The malicious ASes need to be saved into the RCDB so that other third detection method instances can detect the hijacking collaboratively. The network operators, in turn, should ignore and remove the malicious announcements from their routing tables. The third detection method will have the same result efficiency of the second detection method as the third detection method is based on second detection method and both use one collaborative detection method. Therefore, the scenario of the hijacking and detection is not discussed.

6.5 Evaluation and comparing to previous works

Since the detection methods do not concern to the MOAS conflicts, the result of detecting the hijackings will not be affected. However, MOAS conflicts attributes a big issue to previous detection methods [16] [20] [40] but the detection methods took this into account by validating the suspicious ASes to the organisations that could announce their IP prefixes with more than one ASN. In addition, since the detection methods proposed in chapter 4 and 5 are

based on tracing the signature of IP prefix hijacking inside BGP updates and dealing with data (IP prefixes and ASNs) that directly involve in the hijacking, they do not need to look into routing policies (e.g. AS aggregation and IP prefix confederation) engineering. In other words, sub-prefix hijacking will not form any affects to the detection methods to be detected as supper-prefix hijacking. However, previous detection methods did not take routing policy into account; therefore [19] [15] and [32] failed to prevent hijacking. Moreover, the detection methods are only based on IP prefixes and ASNs because the BGP policy of BGP allows for network operators not to send their full data, which makes hijacking detection in some methods is very complicated. Searching for the hijacking in different time-slots of updates is considered the most difficult factor to the second detection method because the hijacking might appear in different time-slots. However, collaborative detection method was proposed in section 6.2 to allow the second detection method instances to work collaboratively to overcome this issue by using sliding window, which proposed in section 6.3.2.

The Controller proposed in section 6.3.2, used to solve difference speed issue between the detection method instances and routers while sliding window used to distribute updates to the detection methods instances. Both techniques work well and can overcome the limitations of the detection methods proposed in chapter 4 and 5. The advantage of the sliding window is to prevent different detection method instances from processing same BGP update time-slot. Another advantage of the collaboration is that the detection methods instances on the servers can distribute the burden and help the detection method instances to find the hijacking very easily. The architecture allows the detection methods to detect IP prefix hijacking quickly because each detection method on the network can check RCDB per specific period of time and then provides the network operator with caught hijackings. The remote centralized database is very useful because it can give other routers pre-knowledge before they receive

the packet and can help routers to avoid spreading malicious packets by removing the fake routes from the routing table.

6.6 Summary

This chapter discusses a collaborative architectural method for linking the proposed detection methods in chapter 4 and 5 to BGP routers and a remote centralised database. The collaborative architecture of the detection methods for detecting hijackings is considered a novel method for securing the BGP. The architecture is composed of detection methods proposed in chapter 4 and 5, a remote centralised database and BGP routers. These methods are linked to BGP routers separately and concurrently to the centralised database. The detection method instances can detect IP prefix hijacking separately and share them with other copies of the detection methods. Finally, BGP router operators are provided with notifications of IP prefix hijackings. The collaborative architecture has some advantages, which can be summarised in in section 6.5. It also increases the accuracy of the detection methods, the speed of detecting IP prefix hijackings, transparency, deployability and integration with BGP routers. The collaborative architecture displays some benefits, such as detecting hijacking quickly and alerting BGP routers of real hijackings.

Chapter : 7 Conclusion

7.1 Achievements

Chapter 1 pointed out to the aims and objectives of the research to detect IP prefix hijacking in the BGP. These aims and objectives were summarised into two things to be achieved: the background of BGP and the hijacking, and the three proposed detection methods, which are based on different techniques: statistical analysis, attack signature, and a suspicious ASes connectivity-behaviour. In terms of the background, the research can study the architecture, the communication, the vulnerabilities, and the security of the BGP and how it works based on the contents of the update messages. In addition, the research investigated the solutions that used to secure the BGP and detect IP prefix hijacking and focused on their advantages and limitations in order to avoid them in the proposed solutions.

First detection method (statistical analysis) aimed to find an indication to the IP prefix hijacking from analysing normal and malicious behaviour of routers during hijacking days (24/02/2008) and normal days (23/02/2008and 25/02/2008) of BGP updates. However, the detection method failed to detect IP prefix hijacking because it did not take into account that normal, potential analytical time-slots of raw BGP updates could contain hijacks, while the other two detection methods have advantages that make them work highly efficiently. The issue with the first detection method is that it does not have a mechanism for separating benign BGP updates from hijacked BGP updates. However, this method can supply the other two detection methods with good analysis and data preparation. The detection method introduced many advantages that might not be directly related to detecting IP prefix hijacking but to the other two detection methods. For example, it provides them with data pre-

processing and organisation. The detection method also performed a deep investigation of the BGP updates in terms of the occurrence and appearance of IP prefix hijacking in the raw data in order to get a general view of the functionality of the BGP and hijacking behaviours. Another advantage of this detection method is that it can gather the previous proposed detection method features and analyse their benefits and relations to the IP prefix hijacking. Moreover, the detection method found a clear mechanism for deciding the size of the processed time-slots of BGP updates so that it is not so big that it could affect the speed of the detection method speed or omit hijacking events. Finally, the detection method shows the importance of finding a good and an accurate way to separate benign BGP updates from malicious updates in the raw data before doing any hijacking detection.

Second detection method, which is based on the signature attack, uses a novel approach to map ASes to their IP prefixes then compare one [AS, IP prefix] to many [ASes, IP prefixes] in one time-slot and validate the results using RIRs. The detection method is based on the Route Views project, which means the BGP updates are collected from real sources, not simulated routers. Since BGP updates are very large, the detection method also needs to use the quickest search algorithm (Binary Search Algorithm, or BSA) while mapping ASes to their IP prefixes, which was achieved in this detection method. In other words, the detection method employed data reduction to delete all repeated ASNs, IP prefixes and routing advertisement operations, which results in detecting hijacks quickly. MOAS conflicts (one organisation with multiple AS numbers) that could show up due to the typical management of distributing ASes and IP prefixes are calculated before making a decision about the results of the suspicious ASes. The detection method also keeps the identity of the hijackers along with the process of detecting the IP prefix hijacking so that it can determine malicious ASes in the future. The second detection method able to distinguish hijacking from other events, as it is

only focus on the signature of hijacking not the instability of the routers that can be done by different reasons. The second detection method can detect hijacking continuously because it is searching for the signature in the updates; unlike prevention methods, when a fake announcement passes the security technique will not be detected till the impersonator withdraws it. In addition, the detection method is not affected by routing policies conflicts such as ASes aggregation because it deals with ASes as separate entity. The detection method can achieve multiple-sampling technique because it uses time-window to decide the time-slot size every specific period of time to search for the hijacking in it. The detection method only achieved half self-checking because it is based on RIRs for validating the detection results that are collected in the SASL. Furthermore, the detection method does not assume that the first collected BGP updates are normal as in the PGBGP, but maps 15-minute BGP updates in order to avoid hijackings that might be embedded in normal behaviour. Periodically, some normal changes occur to inter-domain routing information, such as modifying an organisation's ASNs and assigning a closed organisation's IP prefixes to a new organisation while detecting the IP prefix hijack. Since the history of registered routing information is changing continuously, previous detection methods are unable to avoid these factors and detect IP prefix hijacks accurately. However, because the detection method algorithm is based on active self-checking (collecting and comparing origin ASes and IP prefixes of BGP updates), it works against this kind of unnoticeable factor, which make it very accurate when compared to other detection methods. The second detection method can also detect the hijacking with no false positives because it is resistible to MOAS. The detection methods can achieve deployment because it works far away from the routers' infrastructure specifications. The detection method can be integrated with the BGP without any problem. The detection method can detect the hijacking without affecting the performance of exchanging updates

among routers. Thus, it can achieve router-friendly feature, as there is no need for the detection method to be connected to all routers on the Internet.

Third detection method is combined with the second detection method to receive its outputs as inputs for tracing the behaviour of suspicious ASes in detecting IP prefix hijacking. Namely, the second detection method can be used as a filter for suspicious and purely benign AS behaviour, while the third method judges the behaviour of the suspicious ASes. This third detection method is based on the supervised learning and uses five supervised learning classifiers that suit the dataset characteristics, such as size, class dimensionality, data type and data distribution. The detection method uses supervised learning, which is considered the best and most accurate among machine learning types. The detection method uses many AS BGP updates collected from different regions up to about 50 ASes. New behavioural features and a manual created dataset were proposed in the connective-based method in order to arrive at a good result comparing to other detection solutions. For example, the detection method in chapter 4 reaches a 96% accuracy rate but iSPY can detect the hijacking with a false negative ratio below 45% and a false positive ratio below 17%. Furthermore, the detection method can detect hijacking in a percentage result so it can easily judge the accuracy of the detection, but some previous solutions claim the accuracy without any proof. The method classifies suspicious AS behaviour instead of purely malicious AS behaviour against purely benign behaviour, which means the classification is performed very fast. In other words, the algorithm of the detection method is based on signature attack as a pre-processing method to label the dataset, which is then trained and tested. The detection method extracts the new features from data that are directly related to the hijacking such as ASNs and IP prefixes, not to the stability of routers. This detection method also computes the quality of the proposed features based on the similarities and differences between malicious and benign behavioural

patterns. Moreover, the detection method is based on the information of suspicious ASes in the RIRs, which means it can collect data from an accurate source. The detection method also can confirm the accuracy of the detection method by using relational behaviour of suspicious ASes and split-test option with different supervised learning classifiers to evaluate the accuracy. Another achievement of the detection method is to detect hijacking very quickly by removing all redundant hijackings in the classification dataset, picking classifiers that are considered quicker than classifiers that are based on the regression analysis and only works on suspicious ASes. The detection method is very transparent so it can work and detect IP prefix hijacks automatically and without the requirement that all routers on the Internet be integrated with the programme; rather, some routers do the detection on behalf of others. It can also detect short-lived hijacks because it is based on a decentralised structure of detection (collaboration), as in chapter 6. This collaboration helps the method detect hijacks quickly because some last for a short time [5]. The detection methods can be deployed because it built to work separately from the routers and identifying a way to be integrated with the BGP and does not need any modifications to the infrastructure of current routes. Finally, the detection method can be integrated to the routers without affecting routing performance.

7.2 Difficulties

There is no database for community BGP research that can store benign updates in isolated locations away from malicious updates, so that researchers can use them directly and apply appropriate analyses to them. The lack of this database makes it very difficult for the first detection method to work efficiently; therefore, it failed to detect IP prefix hijacking.

Blocks of AS numbers are not delegated and given sequentially to organisations that use the BGP to exchange routing information with their neighbours; instead, they are given randomly. It would be very difficult to predict the real owner of an IP prefix if the AS numbers of AS-blocks of an organisation are not limited by upper and lower AS numbers. For example, YouTube can have 36040 and 36561 AS numbers but many organisations can have AS numbers in between these two, such as 36041, which belongs to the Savannah College of Art and Design Network Information. This way of distributing AS numbers makes it very hard for researchers to determine the whole AS numbers of a specific organisation so that when it uses multiple AS numbers to announce one IP prefix its AS numbers can easily be predicted. As a result, the second detection method uses RIRs to verify caught suspicious ASes though registered routing information.

IP address blocks are also not distributed sequentially and that leads to difficulties while tracing the IP prefixes that a specific organisation has. As a result, the third detection method focused on the connectivity of the suspicious ASes to extract features from and predict the real owner of a specific IP prefix. Relative similarity between normal behaviour and abnormal behaviour makes it slightly difficult for classifiers to differentiate normal and malicious behaviour during classification.

Changing the ASNs and IP addresses information of organisations occurs continuously in the RIRs and the IANA, which makes labelling benign and malicious behaviours of suspicious ASes very difficult to be classified because some of the organisations are no longer available on the RIR and IANA databases. As a result, any classification database (supervised dataset) will not have a chance to contain big samples of benign and malicious suspicious AS behaviours. None of the RIRs provides extended histories of the incidents and that also

makes it difficult for researchers to have a big sample of benign and malicious suspicious AS behaviours. These three factors make the labelled dataset proposed in section 5.2.1 very small.

Delegating ASNs and IP prefixes happens through the IANA, RIRs and several ISPs, which makes collecting validation information for the verification table very difficult. In other words, ASNs and IP prefixes can be distributed by many organisations, and if there is a need to find the owner of an IP prefix, any detection method has to have all databases of the delegators in addition to the IANA and RIR databases.

7.3 Limitations

The limitation of first detection method is that it does not have a mechanism that can differentiate benign BGP updates from BGP malicious updates before doing analysis. The lack of this mechanism means that the detection method is not able to detect IP prefix hijacking. Second detection method has a serious limitation, which it cannot detect the hijackings when they are announced in different time-slots. The limitation of the third detection method is that it still yields about 3% false positives and 2% false negatives. It also requires all routers to advertise their own IP prefixes, at least once, during their exchanging routing information so the detection method can have enough historical or prior knowledge of the ownership of the IP prefixes. However, collaborative detection method is proposed to overcome the limitations of the proposed second and third detection methods. The collaborative detection method can evaluate the final accuracy of the second and third detection methods when they are integrated with the routers. Another advantage of the collaborative detection method is that each instance of the detection methods are installed on

the router can only check its own received BGP update packets from the Controller. Thus, there is no load to the proposed detection method algorithms to find the hijacking. Another advantage of this collaboration is checking hijacks signatures when occur periodically and in different time-slots. For example, if router A cannot detect the hijack at 1:15 AM because it does not have the time slot to do the check, there will be other copies of the algorithm integrated with the neighbouring routers to perform the hijacking check.

7.4 Future work

The detection method proposed in chapter 4 is considered is half self-checking because it depends on the RIRs for validating the results. The detection method needs to be improved and be fully self-checking. In other words, the detection method has to be totally depends on the BGP updates. For examples, the detection method can monitor the changes of each AS in the update itself such as number of fixed neighbours and the away showing and hiding some attribute in order to decide the owner of the IP prefix. In the same way, the detection method in chapter 5 still has 6% false positives and false negatives there it needs to be enhanced in future. Using ANN can be another a good choice to retest the third detection method. For collaborative detection method, there must be a flexible way for removing the hijacking and reducing the spread of the hijacking without need to send notifications to the network operators. This approach could be achieved by embedding a new attribute into BGP. This improvement will help to take the responsibly out of the network operators with regarding to dealing with the hijackings and make the detection fully based on the methods.

References

[1] "Understanding the Internet's insecure routing infrastructure | Ars Technica." [Online]. Available: http://arstechnica.com/tech-policy/2010/11/understanding-the-Internets-insecure-routing-infrastructure/.

[2] K. Butler and P. Mcdaniel, "A Survey of BGP Security," vol. V, no. April, pp. 1–35, 2005.

[3] S. Goldberg, "Why is it taking so long to secure internet routing?," *Commun. ACM*, vol. 57, no. 10, pp. 56–63, Sep. 2014.

[4] J. Schlamp, J. Gustafsson, M. Wählisch, T. C. Schmidt, and G. Carle, "The Abandoned Side of the Internet: Hijacking Internet Resources When Domain Names Expire," in *arXiv preprint arXiv: …*, 2015, pp. 188–201.

[5] P. Vervier, O. Thonnard, and M. Dacier, "Mind Your Blocks: On the Stealthiness of Malicious BGP Hijacks," in *Proceedings 2015 Network and Distributed System Security Symposium*, 2015, no. February, pp. 8–11.

[6] G. Valadon and N. Vivet, "Detecting BGP hijacks in 2014 BGP Hijacking for Cryptocurrency Profit Reported by Dell SecureWorks on August 7 2014," 2014. [Online]. Available: http://www.nosuchcon.org/talks/2014/D3_04_Guillaume_Valadon_Nicolas_Vivet_detecting_BGP_hijacks.pdf.

[7] I. O. de Urbina Cazenave, E. Kosluk, and M. C. Ganiz, "An anomaly detection

framework for BGP," in *2011 International Symposium on Innovations in Intelligent Systems and Applications*, 2011, pp. 107–111.

[8] J. D. Gardiner, "Multiple Markov Models for Detecting Internet Anomalies from BGP Data," in *2009 DoD High Performance Computing Modernization Program Users Group Conference*, 2009, pp. 374–377.

[9] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 582–592, 2000.

[10] M. Wubbeling, T. Elsner, and M. Meier, "Inter-AS routing anomalies: Improved detection and classification," in *2014 6th International Conference On Cyber Conflict (CyCon 2014)*, 2014, pp. 223–238.

[11] M. Lepinski, S. Kent, and D. Kong, "A Profile for Route Origin Authorizations (ROAs)." Request for Comments: 6482, pp. 1–9, 2012.

[12] M. Wählisch, O. Maennel, and T. C. Schmidt, "Towards detecting BGP route hijacking using the RPKI," in *ACM SIGCOMM Computer Communication Review*, 24-Sep-2012, vol. 42, no. 4, p. 103.

[13] E. Kranakis, P. C. Oorschot, T. Wan, P. van Oorschot, and T. Wan, "Security Issues in the Border Gateway Protocol (BGP)," 2005.

[14] "Securing BGP through Secure Origin BGP (soBGP)." [Online]. Available: ftp://ftp-eng.cisco.com/sobgp/presentations/BCR-soBGP.pdf. [Accessed: 02-Jan-2013].

[15] B. W. <bew@cisco.com>, “Secure Origin BGP (soBGP) Certificates.” [Online]. Available: http://tools.ietf.org/html/draft-weis-sobgp-certificates-01. [Accessed: 06-Dec-2013].

[16] H. Cao, M. Wang, X. Wang, and P. Zhu, “A Packet-Based Anomaly Detection Model for Inter-domain Routing,” in *2009 IEEE International Conference on Networking, Architecture, and Storage*, 2009, pp. 192–195.

[17] H. Balakrishnan, *How YouTube was “ Hijacked ,”* no. May. 2009.

[18] C. D. Marsan, “Six worst Internet routing attacks.” [Online]. Available: http://www.networkworld.com/article/2272520/lan-wan/six-worst-internet-routing-attacks.html. [Accessed: 28-Jan-2014].

[19] R. Holloway, “Interdomain Routing Security (BGP-4) A Comparison between S-BGP and soBGP,” no. February, pp. 1–79, 2009.

[20] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, “Ispy:Detecting IP Prefix Hijacking on My Own,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, p. 327, Oct. 2008.

[21] D. Meyer, “Index of bgpdata,” *University of Oregon*, 2003. [Online]. Available: http://archive.routeviews.org/bgpdata/. [Accessed: 11-Nov-2013].

[22] C. M. Lewandowski, “A Border Gateway Protocol 4 (BGP-4).” Cambridge University Press.

[23] A. Zeb and M. Farooq, “BGP Threats and Practical Security,” no. March, 2011.

[24] D. Ward, J. Scudder, R. Bush, and R. Austein, “BGP Prefix Origin Validation,” 2013.

[25] Todd Underwood, “Con-Ed Steals the ‘Net,” 2006. [Online]. Available: http://research.dyn.com/2006/01/coned-steals-the-net/. [Accessed: 15-Mar-2013].

[26] S. White, “Pakistan move knocked out YouTube - CNN.com.” [Online]. Available: http://bookchin.net/trip-research/youtube/pakistanBlocksYoutube.html. [Accessed: 28-Jan-2014].

[27] NEWS ABC, “Pakistan lifts YouTube ban.” [Online]. Available: http://www.abc.net.au/news/2008-02-27/pakistan-lifts-youtube-ban/1054918?section=world. [Accessed: 28-Jan-2014].

[28] Andree Toonk, “Chinese ISP hijacks the Internet,” 2010. [Online]. Available: http://www.bgpmon.net/chinese-isp-hijacked-10-of-the-internet/. [Accessed: 01-Feb-2013].

[29] Pat Litke and Joe Stewart, “BGP Hijacking for Cryptocurrency Profit,” 2014. [Online]. Available: https://www.secureworks.com/research/bgp-hijacking-for-cryptocurrency-profit. [Accessed: 03-Dec-2014].

[30] Cisco, “BGP Case Studies,” 2008. [Online]. Available: http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/26634-bgp-toc.html. [Accessed: 15-Feb-2013].

[31] W. Aiello, F. Park, J. Ioannidis, F. Park, P. Mcdaniel, and F. Park, “Origin Authentication in Interdomain Routing,” 2003.

[32] T. Wan, E. Kranakis, and P. C. Van Oorschot, “Pretty Secure BGP (psBGP),” *12th Annu. Netw. Distrib. Syst. Secur. Symp.*, no. February, pp. 1–23, 2005.

[33] N. S. Series, *Aspects of Network and Information Security.pdf*. IOS Press, 2008.

[34] G. Michaelson, “Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs),” 2012.

[35] P. C. Van Oorschot, T. Wan, and E. Kranakis, “On interdomain routing security and pretty secure BGP (psBGP),” *ACM Trans. Inf. Syst. Secur.*, vol. 10, no. 3, p. 11–es, Jul. 2007.

[36] P. McDaniel, W. Aiello, K. Butler, and J. Ioannidis, “Origin authentication in interdomain routing,” *Comput. Networks*, vol. 50, no. 16, pp. 2953–2980, Nov. 2006.

[37] M. Thottan and C. Ji, “Anomaly Detection in IP Networks,” vol. 51, no. 8, pp. 2191–2204, 2003.

[38] P.-A. Vervier *et al.*, “Malicious BGP hijacks: Appearances can be deceiving,” in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 884–889.

[39] M. Lad, D. Massey, D. Pei, and Y. Wu, “PHAS: a prefix hijack alert system,” *Proc. USENIX Secur. ...*, pp. 153–166, 2006.

[40] J. Karlin, S. Forrest, and J. Rexford, “Pretty Good BGP: Improving BGP by Cautiously Adopting Routes,” in *Proceedings of the 2006 IEEE International Conference on Network Protocols*, 2006, pp. 290–299.

[41] IANA, “Number Resources,” 2012. [Online]. Available: http://www.iana.org/numbers. [Accessed: 01-Jan-2012].

[42] M. N. Inc, “Overview of the IRR.” [Online]. Available: http://www.irr.net/docs/overview.html. [Accessed: 05-Jul-2012].

[43] G. Siganos and M. Faloulsos, “Analyzing BGP policies: methodology and tool,” in *IEEE INFOCOM 2004*, 2004, vol. 3, pp. 1640–1651.

[44] K. Sriram, O. Borchert, O. Kim, P. Gleichmann, and D. Montgomery, “A Comparative Analysis of BGP Anomaly Detection and Robustness Algorithms,” in *2009 Cybersecurity Applications & Technology Conference for Homeland Security*, 2009, pp. 25–38.

[45] L. Yuan, “BGP Anomaly Detection Using Wavelet Analysis,” vol. 273, no. March, 2004.

[46] N. Al-Rousan, S. Haeri, and L. Trajkovic, “Feature selection for classification of BGP anomalies using Bayesian models,” in *2012 International Conference on Machine Learning and Cybernetics*, 2012, pp. 140–147.

[47] J. Mai, L. Yuan, and C.-N. Chuah, “Detecting BGP anomalies with wavelet,” in

NOMS 2008 - 2008 IEEE Network Operations and Management Symposium, 2008, pp. 465–472.

[48] J. Qiu, L. Gao, S. Ranjan, and A. Nucci, "Detecting bogus BGP route information: Going beyond prefix hijacking," in *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*, 2007, vol. 1, pp. 381–390.

[49] M. Thottan, G. Liu, and C. Ji, "Anomaly detection approaches for communication networks," *Networks*, vol. 2, pp. 1–19, 2010.

[50] RIPE, "RIS Raw Data — RIPE Network Coordination Centre." [Online]. Available: https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data. [Accessed: 15-Nov-2012].

[51] MangoB2B, "Zebra – multi-server routing software." [Online]. Available: http://www.zebra.org/. [Accessed: 20-Feb-2017].

[52] "Pakistan hijacks YouTube - Renesys." [Online]. Available: http://www.renesys.com/2008/02/pakistan-hijacks-youtube-1/.

[53] SlidePlayer, "Towards an Accurate AS-level Traceroute Tool." [Online]. Available: http://slideplayer.com/slide/8131767/. [Accessed: 02-May-2013].

[54] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz, "Towards an accurate AS-level traceroute tool," in *Proceedings of the 2003 conference on Applications, technologies,*

architectures, and protocols for computer communications - SIGCOMM '03, 2003, p. 365.

[55] V. Menon and W. M. Pottenger, "A Higher Order Collective Classifier for detecting and classifying network events," in *2009 IEEE International Conference on Intelligence and Security Informatics*, 2009, pp. 125–130.

[56] SolarWinds, "What is NetFlow by SolarWinds." [Online]. Available: http://www.solarwinds.com/what-is-netflow. [Accessed: 20-Feb-2017].

[57] T. Qiu, L. Ji, D. Pei, J. Wang, J. (Jim) Xu, and H. Ballani, "Locating Prefix Hijackers using LOCK," *USENIX Secur. Symp.*, vol. 9, no. 6, pp. 135–150, 2009.

[58] RIPE NCC, "Routing Information Service." [Online]. Available: http://www.ripe.net/ris/. [Accessed: 05-Oct-2013].

[59] N. M. Al-Rousan and L. Trajkovic, "Machine learning models for classification of BGP anomalies," in *2012 IEEE 13th International Conference on High Performance Switching and Routing*, 2012, pp. 103–108.

[60] S.-C. Hong, J. W.-K. Hong, and H. Ju, "IP prefix hijacking detection using the collection of as characteristics," *2011 13th Asia-Pacific Netw. Oper. Manag. Symp.*, no. SEPTEMBER, pp. 1–7, 2011.

[61] J. Zhao and Y. Wen, "Evaluation on the influence of internet prefix hijacking events," *Comput. Sci. Inf. Syst.*, vol. 10, no. 2, pp. 611–631, 2013.

[62] J. Li, D. Dou, Z. Wu, S. Kim, and V. Agarwal, “An internet routing forensics framework for discovering rules of abnormal BGP events,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 3, p. 55, Oct. 2005.

[63] T. Qiu, L. Ji, D. Pei, J. Wang, J. Xu, and H. Ballani, “LOCK: Locating Countermeasure-Capable Prefix Hijackers,” 2008.

[64] E. Bijnen and M. Magiel, “System and Network Engineering Research Project 1 Detecting IP Hijacking Through Server Fingerprinting,” 2014.

[65] X. Hu and Z. M. Mao, “Accurate Real-time Identification of IP Prefix Hijacking,” in *2007 IEEE Symposium on Security and Privacy (SP ’07)*, 2007, no. 2, pp. 3–17.

[66] D. Meyer, “University of Oregon Route Views Archive Project,” *University of Oregon*. [Online]. Available: http://archive.routeviews.org/bgpdata/2008.02/UPDATES/. [Accessed: 05-Oct-2013].

[67] “bgptools,” *MIT University*. [Online]. Available: http://nms.lcs.mit.edu/software/bgp/bgptools/. [Accessed: 10-Oct-2013].

[68] S. Deshpande, M. Thottan, T. K. Ho, and B. Sikdar, “A Statistical Approach to Anomaly Detection in Interdomain Routing,” in *2006 3rd International Conference on Broadband Communications, Networks and Systems*, 2006, pp. 1–10.

[69] John Stamatakis, “Pen Test Live: Download Database,” 2014. [Online]. Available: http://www.pentestlive.com. [Accessed: 09-Jan-2014].

[70] MathWork, “Cell Arrays - MATLAB & Simulink - MathWorks United Kingdom.” [Online]. Available: http://uk.mathworks.com/help/matlab/cell-arrays.html. [Accessed: 12-Jul-2014].

[71] A. Dalal, “Searching and Sorting Algorithms,” 2004. [Online]. Available: http://www.cs.carleton.edu/faculty/adalal/teaching/f04/117/notes/searchSort.pdf. [Accessed: 08-Dec-2014].

[72] ARIN (American Registry for Internet Numbers), “Whois-RWS.” [Online]. Available: http://whois.arin.net/ui/query.do. [Accessed: 01-Feb-2014].

[73] A. (Asia P. N. I. Centre), “APNIC - Query the APNIC Whois Database.” [Online]. Available: http://wq.apnic.net/apnic-bin/whois.pl.

[74] RIPE, “RIPE Database Query,” 2013. [Online]. Available: https://apps.db.ripe.net/search/query.html. [Accessed: 13-Mar-2014].

[75] J. Zhang, J. Rexford, and J. Feigenbaum, “Learning-based anomaly detection in BGP updates,” in *Proceeding of the 2005 ACM SIGCOMM workshop on Mining network data - MineNet ’05*, 2005, p. 219.

[76] MathWorks, “Network Analysis and Visualisation,” 2015. [Online]. Available: http://uk.mathworks.com/help/bioinfo/network-analysis-and-visualization.html. [Accessed: 10-Nov-2015].

[77] X. Wu *et al.*, “Top 10 algorithms in data mining,” *Knowl. Inf. Syst.*, vol. 14, no. 1, pp.

1–37, Jan. 2008.

[78] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, “Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?,” *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, 2014.

[79] G. Kaur and A. Chhabra, “Improved J48 Classification Algorithm for the Prediction of Diabetes,” *Int. J. Comput. Appl.*, vol. 98, no. 22, pp. 13–17, 2014.

Appendix

Posters

Detecting IP prefix Hijacks

Hussain Alshamrani
Centre of Security, Communications and Network Research, Plymouth University , UK

Abstract

Border gateway protocol (BGP4) has significant security issues regarding ASes and IP prefixes, such as impersonating the ownership of other AS IP prefixes. There is a variety of research methods already used to secure BGP4 such as using historical-based and statistical model-based; in addition, recent research has already investigated IP prefix hijacking, but accuracy, robustness, and efficiency of proposed methods are still low. The algorithm in this poster detects IP prefix hijacks by monitoring the behaviour of BGP4 edge routers. The algorithm aims to find IP prefix hijacks of ASes in same and different regions (national and international). From a timing perspective, IP prefix hijack incidents should be detected within 15 minutes of their occurrence, based on the fact that the effect of invalid routes has to be reduced towards the impersonated organisation.

Existing Solutions and their Limitations

- Rule-/Packet-Based - not real-time detection [2].
- Routing Table-Based – unable to access ISPs' routing table information because they are not allowed to reveal some companies' commercial policies [2].
- Address Delegation and Origin Authentication – require high modification the router infrastructure and expensive computation [3].
- IA (Identity Assertion) e.g. Prefix Assertion List – cannot distinguish routers' instability of normal events such as power cut-off from hijacks [5].
- Historical-Based – cannot differentiate events from valid policy changes [1].
- Registry-Based – has high false positives and negatives [4].
- Combination of Historical and Registry-Based
- Statistical-Based (e.g. HMM, SVM and Naïve Bayesian)

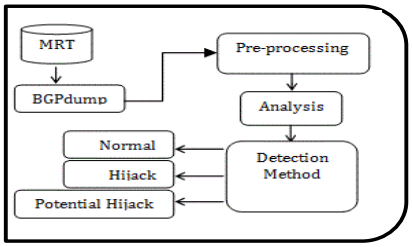
Problem

- IP prefix hijack can affect router stability and network availability.
- February 2008, Pakistan telecomm blocked YouTube unintentionally.
- Google was blocked on the 7th May 2005.
- ISP in Malaysia blocked Yahoo on 2004.
- The impact of this attack is that the IP prefix impersonator can compromise the direction of the packets.

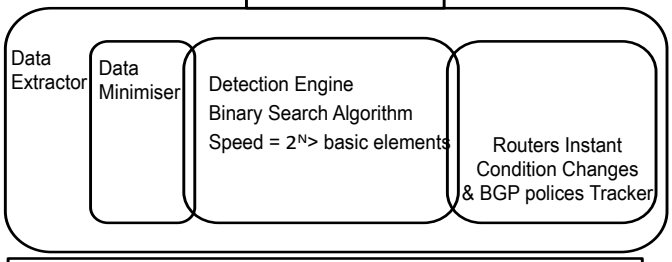
Solution

- Intelligent data analysis – pre-process BGP updates and evaluate statistical features

Architecture



Detection Method



Data Extractor:
- In charge to extract relative data per 15 minutes
- Pre-process data and put them in analytical way

Data Minimiser:
- Responsible to reduce the number of data

Detection Engine:
- Responsible to map AS origin to the real owner
- Detect IP prefix hijack based on BGP updates anomalies

Routers Instant Condition Changes & BGP polices Tracker:
- Verifying the IP prefix hijacks found

Conclusion

Security in BGP4 was extensively investigated in order to protect the integrity of BGP and detect or prevent illegitimate events. However, the BGP still suffers form some vulnerabilities and subject to a serious attack, which is IP prefix hijacking. Different techniques and methods were proposed to detect this attack accurately. Nonetheless, some of them have inherent limitations. In the summarised research project, detection method consists of four components ; each one has a specific task. The proposed detection method aims to link intelligent data analysis and BGP polices to detect and reduce the impact of IP prefix hijacks.

References

[1] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, L. Zhang, "An Analysis of BGP Multiple Origin AS (MOAS) Conflicts," 1930.
[2] H. Cao, M. Wang, X. Wang, and P. Zhu, "A Packet-Based Anomaly Detection Model for Inter-domain Routing," *2009 IEEE Int. Conf. Networking, Archit. Storage*, pp. 192–195, Jul. 2009.
[3] A. Zeb and M. Farooq, "BGP Threats and Practical Security," no. March, 2011.
[4] I. O. de Urbina Cazenave, E. Kosluk, and M. C. Ganiz, "An anomaly detection framework for BGP," *2011 Int. Symp. Innov. Intell. Syst. Appl.*, pp. 107–111, Jun. 2011.
[5] M. Wählisch and T. C. Schmidt, "Towards Detecting BGP Route Hijacking using the RPKI," no. ii, pp. 103–104, 2012.

Papers

Detecting IP prefix hijacking using data reduction-based and binary search algorithm

Hussain Alshamrani
Centre for security, Communications and Network Research (CSCAN)
Plymouth University
Plymouth, UK
hussain.alshamrani@plymouth.ac.uk

Bogdan Ghita
Centre for security, Communications and Network Research (CSCAN)
Plymouth University
Plymouth, UK
bogdan.ghita@plymouth.ac.uk

David Lancaster
Centre for security, Communications and Network Research (CSCAN)
Plymouth University
Plymouth, UK
david.lancaster@plymouth.ac.uk

Abstract—In spite of significant on going research, the Border gateway protocol (BGP) still encompasses conceptual vulnerability issues regarding impersonating the ownership of IP prefixes for ASes (Autonomous Systems). In this context, a number of research studies focused on securing BGP through historical-based and statistical-based behavioural models. This paper proposes a novel algorithm aiming to track the behaviour of BGP edge routers and detect IP prefix hijacks based on a typical signature. The algorithm parses the BGP advertisements in order to detect the apparent relocation of specific IP prefixes, either in the same or in different regions. The algorithm aims to identify IP prefixes by multiple independent ASes. The method differs from routing consistency monitoring, which faces difficulties detecting events at the edge of the BGP infrastructure. Based on the RIRs' database, the algorithm can detect national

and cross-border IP prefix hijacks very quick. However, 5 results out of 16 were not accurate therefore the algorithm has some false positives and needs further improvement to be done in future.

Keywords—BGP advertisements; Binary Search Algorithm; Data Reduction; IP prefix; origin AS

I. Introduction

BGP remains the protocol of choice for core Internet interconnectivity. At its core, the protocol consists of four messages: OPEN and KEEPALIVE (both used for session establishment and connection control), NOTIFICATION (to inform peers of errors), and UPDATE (to build and update routing tables) [1]. From the previous studies, some researchers tried to detect IP prefix hijacks based on monitoring routers' stabilities. Nonetheless, their methods could not reliably distinguish IP prefix hijacks from normal events, such as power cut-off and submarine cuts [2]. In addition, RPKI (Resource Publication Infrastructure) system took place to detect BGP route hijacking, however the system had several false positives and negatives and needs further refinements [3]. Lastly, some methods analyse routing tables (table-based) in order to detect IP prefix hijacks, but these methods have two serious issues: organisations may refuse to provide their routing tables as well as not being able to detect the hijack on time [4]. This paper focuses on vulnerabilities caused by the implicit trust between BGP peers when receiving UPDATE messages. A novel method of detecting IP prefix hijacking incidents, based on data reduction and Binary Search Algorithm, is built in order to accurately and timely detect IP prefix hijacking events. The detection method is

based on an algorithm traces origin ASes and their actual IP prefixes in 15 minute timeslots. The method is designed and tested using UPDATE messages of different routers downloaded from the Route Views Archive Project [5]. As a case study, UPDATE messages were collected from 24th of February 2008 when Pakistan Telecom intended to restrict local access to YouTube, but the advertised UPDATE messages blocked access to YouTube [6] for approximately two hours [7].

The paper provides in section II an overview of a typical IP prefix hijacking incident, its impact on the end users and the ability to observe it, then a description of a specific incident that was used to build the proposed detection method. Section III includes information on data source, data pre-processing, data analysis and the algorithm while section IV discusses findings, some new incidents and the algorithm challenges. Section V introduces the architecture of the detection system, which relies on the collaboration among routers to improve detection efficiency. The paper finishes with conclusions and future work in section VI.

II. IP prefix HIJACKING

This section discusses the process view of the IP prefix hijacking, the impact on the end user, a case study of Pakistan and the YouTube IP prefix hijack and finally the data analysis of the hijack.

A. The Process

IP prefix hijacking occurs when more than one AS announces an IP super-prefix or a sub-prefix that is owned by another AS. However, hijack events could turn up during normal operations of the BGP such as AS confederation and complicated ASN

(Autonomous System Number) changes of organisations. Figure 1 consists of seven ASes; edge router in AS100 represents the real owner (the announcer) of prefix 1.1.1.1.0/24, AS300 aims to hijack the 1.1.1.1.0/24 and the remaining edge routers of AS400, AS5000, AS1000, AS4000 and AS10000 work as propagators to 1.1.1.0/24. The edge router on the AS100 announces the IP prefix 1.1.1.0/24 to an edge router located on AS5000 that, in turn, propagates it to AS1000. There is no direct connection between AS300 and AS100. AS300 could announce the same IP prefix to AS5000 and AS4000 whether before or after AS100. Although both AS100 and AS300 announce the same IP prefix 1.1.1.0/24 to AS5000, this AS cannot detect the hijack but would probably spread it out to AS1000. In addition, AS300 might announce the same prefix, which it does not originally own, to AS4000 then further to other national or international ASes. Moreover, AS300 can announce IP prefix 1.1.1.0/24 indirectly to AS400, which would perhaps announce it to the real origin AS (AS100). However, the real owner will not detect that it had been hijacked because BGP lacks origin authentication. In this case, some edge routers would assume that AS300 could take it to 1.1.1.0/24 in a shorter way shorter than AS100. Consequently, edge routers would withdraw their routes in order to go through AS300.

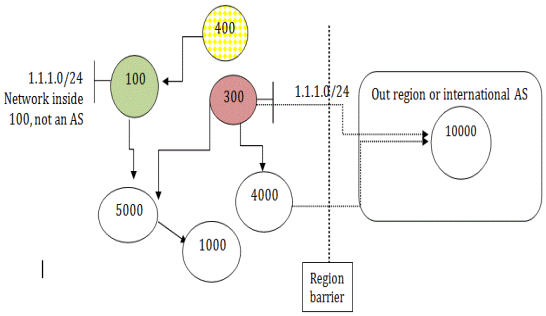


Fig. 1. Process of the IP prefix hijacking

If the case is applied to an extremely popular organisation such as YouTube,

Google and Yahoo, denial of service will have an immediate impact on end users.

B. The Case Study

Below data raw represent snapshots taken between 12:07:00, 24 February 2008 and 12:13:07, 15 February 2009, when Pakistan Telecom erroneously announced one of the YouTube IP prefixes. Packet one in figure 2 shows the first occurrence of the hijack when an edge router belonging to the AS17557 announced the 208.65.153.0/24. However, figure 3 shows a different AS (36561) announcing the same IP prefix. Figure 4 shows the last period of the hijacking activity when the fake route withdrawn by AS17787.

Based on the known hijacked IP prefix in the YouTube incident, a shell script was written to search for fake routes in all of the divided UPDATE messages in the day of the event. The program already knows the ASN of the YouTube and Pakistan Telecommunication. The program found that the hijacked IP prefix showed up in 5 quarters 74, 79, 81, 82 and 83. The impersonator (AS17557) started by announcing 208.65.153.0/24 (in the 74th quarter) before the legitimate owner. The legitimate AS (36561) began to announce the same IP prefix in 79th and 81st quarter but in the absence of the hijacker. The impersonated IP prefix again started to appear in the 82nd and 83rd quarter but under two different origin ASes. The hijacked IP prefix was withdrawn by an AS (AS17787), which had a direct link to the impersonator.

```
TIME: 02/24/08 18:47:57
TYPE: BGP4MP/MESSAGE/Update
FROM: 149.20.65.198 AS1280
TO: 128.223.51.102 AS6447
ORIGIN: IGP
ASPATH: 1280 6461 3491 17557
NEXT_HOP: 149.20.65.198
MULTI_EXIT_DISC: 30
ANNOUNCE
208.65.153.0/24
```

Fig. 2. Raw data of YouTube hijacking when it started

```
TIME: 02/24/08 20:51:31
TYPE: BGP4MP/MESSAGE/Update
FROM: 202.232.0.3 AS2497
TO: 128.223.51.102 AS6447
ORIGIN: IGP
ASPATH: 2497 3549 36561
NEXT_HOP: 202.232.0.3
ANNOUNCE
 208.65.153.0/24
```

Fig. 3. Raw data of YouTube hijacking in the middle

```
TIME: 02/24/08 21:01:21
TYPE: BGP4MP/MESSAGE/Update
FROM: 81.209.156.1 AS13237
TO: 128.223.51.102 AS6447
ORIGIN: IGP
ASPATH: 13237 702 17557 17787
NEXT_HOP: 81.209.156.1
COMMUNITY: 13237:40044 13237:46441
WITHDRAW
 208.65.153.0/24
 203.92.5.0/24
 203.92.4.0/24
ANNOUNCE
 203.215.170.0/24
```

Fig. 4. Raw data of YouTube hijacking at the end

This case study is used to investigate raw data and the BGP messaging footprint of an IP prefix hijacking in order to build a reliable detection method to detect new IP prefix hijacks.

C. IP prefix Hijack Analysis Based on the Case Study

The above snapshots (figure 2, 3 and 4) are analysed according to two factors: extracting data that are directly related to the announcer (last ASN) and relevant to the hijack, then normalising the variable length of ASPATH and ANNOUNCE. Since the routes in the collected updates are generated by the last AS in the ASPATH attribute, pre-processing of messages requires firstly a function to extract the origin ASes from the UPDATE messages. As a result, for a given message, each IP prefix can be associated with its announcing AS and the ASPATH length. Ultimately, this association, in conjunction with the timing of the message, has to be at the core of an IP hijacking detection method, as it provides all the information about which IP prefixes are apparently owned by their announcing AS routers.

ASes and IP prefixes are chosen to be in the last column of the processed data

(dataset) in order to be stabilised automatically by MatLab and to meet its rules. All IP prefixes are converted into integer IP addresses to simplify the loading of data into MatLab, as integer data can be sorted faster than string and to be homogeneous with the rest of the numeric data in the dataset. The stability of data is not a very serious issue for the detection method except they need to be acceptable and loaded in an analytical environment. The most important factor is extracting the same features of all different edge routers.

III. Detection method

Based on the process described in section II, we propose a new detection method in order to detect different remote ASes that probable announce the same IP prefix, and flag these events as potential IP prefix hijacking. The method consists of three main blocks, data pre-processing, data analysis, and detection algorithm. The data pre-processing uses raw BGP announcements as inputs and organises them in CSV (Comma-Separated Values) format. The analysis extracts features, excludes redundant data and provides a unified view of repeated advertised prefixes and associated ASes. The algorithm reduces data dimensions, parses the output of data analysis and identifies any announcements of the same prefix by multiple ASes.

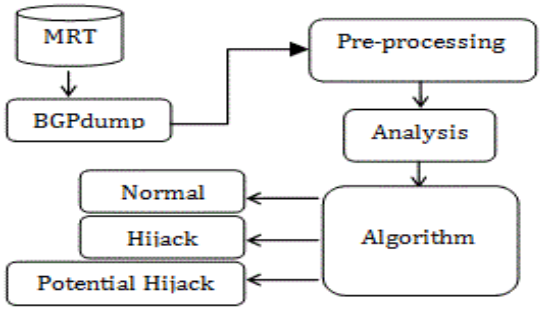


Fig. 5. Data processing and detection method

The detection method is structured as shown in figure 5. The algorithm has a

single output with three values: normal event, certain hijack or potential hijack and includes three tasks, focusing on data mapping, data reduction, and binary search. The remainder of this section will describe each stage of the process in detail.

A. Data Pre-processing

The proposed method uses as input BGP UPDATE messages as seen by a BGP router connected to the network, which are converted to ASCII using BGPdump [8] with some modifications to suit the follow-up analysis. BGPdump is an open source tool used to convert binary data (raw data) into ASCII, as shown in figure 2, 3 and 4 in section II.B. The customised version organises ASCII update packets in several rows and equal columns. ASPATH and ANNOUNCE attributes are not consistent which is likely to make it very complicated to deal with data and trace fake impersonations of routes. As a result, data pre-processing comes as a second step of the detection method in order to perform several tasks such as making raw data organised and consistent. Raw data (ASPATH and ANNOUNCE) have a variable number of the ASes and IP prefixes, which require making them consistent (such as padding variable length fields). As a task for the pre-processing phase the IP prefixes are converted from string IP address into integer IP address in order to unify data type and facilitate detection operation of the algorithm.

B. Data Analysis

When raw data were processed, features are extracted and saved in two different datasets; the first one includes the origin ASes in the last column of the processed data and the second one

includes the IP prefixes with their prefix ranges in the last column as well. The first dataset also has ASPATH length while the second dataset has the announce length and the prefix range. These two datasets were created to map each origin AS with its IP prefixes as it is described in the following subsection. As part of the analysis, the dataset is analysed in 15-minute snapshots, including all the BGP UPDATE messages sent during that period. The algorithm receives each 15-minute snapshot automatically.

C. Algorithm

The algorithm stage has three objectives – firstly to associate the announcer (origin AS) with each advertised IP prefix, secondly to remove duplications of associated origin ASes and IP prefixes and finally to identify any IP prefixes that were announced by more than one AS. The algorithm receives the origin ASes and their IP prefixes, from two different datasets, every fifteen minutes. Origin AS with its extracted features in the first dataset and IP prefixes with its extracted features in the second datasets are loaded dynamically into two different large matrices. Origin ASes, in the first dataset, and their IP prefixes, in the second dataset, are mapped automatically. In order to optimise search for the association between IP prefixes and their corresponding origins, these data are converted into a MatLab cell array [9]. Repetitions of ASNs and IP prefixes are deleted separately. In other words, both columns origin ASes and IP prefixes in the cell array are subjected to data reduction.

Data reduction is applied to the dataset by removing unwanted origin ASNs and duplicate IP prefixes. For example, matrix size before data reduction in quarter 82 is

21968 (rows) by 507 (columns). After reduction, the dimension of mapping cell array becomes 582 by 2. Each 15-minute snapshot compacts multiple announcements for a specific AS into a single row of cell array. After computing unrepeated origin ASes and their unrepeated IP prefixes, the algorithm starts to compare the IP prefix of each AS to the IP prefixes of all the origin ASes in the entire fifteen minutes to find any IP prefixes advertised by multiple ASes.

The comparison algorithm considers the first row of the origin AS and its IP prefixes in the Mapping cell array as a main row column vector. This vector is compared to other origin ASes' IP prefixes vectors. The main vector uses a relatively fast comparison search algorithm (Binary Search Algorithm [10]) to compare the IP prefix of the current origin AS to the remaining origin ASes IP prefixes. The main column vector will be removed from the comparison and the next column vector will be the main vector to be compared to the following IP prefixes of the origin ASes. The lower the number of ASes and announced IP prefixes, the faster the algorithm will be. The processing continues until the end of the cell array. The algorithm separates each detected event with a row vector to differentiate new anomalous cases (detected hijacks). Figure 6 depicts three subscribed components to discover the IP prefix hijacking.

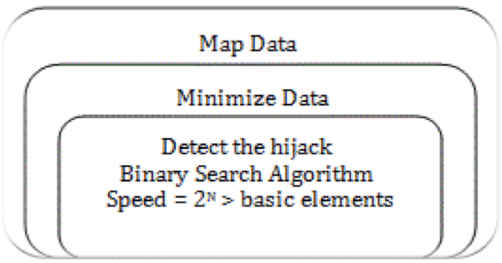


Fig. 6. The organisation of detection method

After converting the IP prefixes to integers in the subsection III.A and sorting

the dataset produced by the reduction phase, the BSA (Binary Search Algorithm) is employed to determine whether more than one AS announces a specific IP prefix. The reason for using BSA is that it executes array comparisons exponentially faster than linear search algorithm (LSA) [10].

TABLE 1. Example of mapping cell array in quarter 82

Order	Origin ASes	IP prefixes
3	137	369760021
4	151	369760021; 369760023; 3697600524
5	174	139438524; 244296124
582	44408	80045022

Table 1 shows the format of the Mapping cell array for two different styles: single IP prefix and multiple IP prefixes, one origin AS can have one or multiple values. The BSA is applied to the origin AS that has more than one value. BSA either ignores normal cases, or catches hijacks and suspicious hijacks. In other words, the detection method has three outputs normal data, potential and certain hijacks.

IV. VALIDATION

The proposed detection method is tested against the dataset that contains the IP prefix hijacking event described in section III.B. The global routing information can be reconstructed using the Route Views Project [11] which includes a comprehensive archive of BGP UPDATE messages. The dataset was generated using BGP UPDATE traces from the Route Views archive. This section demonstrates detection method findings, incidents and algorithm challenges.

A. Findings

ASLoc and IPPLoc in tables 2-6 represent the ASN location and the IP prefix location where the suspicious hijacks were found in the cell array.

However, the third column represents the IP prefixes and their range of super and sub-prefixes. Each row in table 2-6 displays two different origin ASes claiming one IP prefix.

TABLE 2. The incidents captured by the algorithm in Quarter 74

Quarter 74 (starting time of the hijack)				
ASLoc:IPPLoc		ASNs		IP prefix hijacked
18:3	105:4	637	5963	214.15.201.0/24
57:2	424:2	3602	18638	209.5.171.0/24
213:2	377:2	9498	17443	202.140.63.0/24
446:2	642:2	19750	32004	207.181.144.0/24
452:4	492:2	20214	22909	64.139.74.0/24
507:7	725:2	23694	38513	202.87.191.0/24

TABLE 3. The incidents captured by the algorithm in Quarter 79

Quarter 79 without repetition events				
ASLoc:IPPLoc		ASNs		IP prefix hijacked
170:2	489:2	10461	35931	65.171.224.0/22

TABLE 4. The incidents captured by the algorithm in Quarter 81

Quarter 81 without repetition events				
ASLoc:IPPLoc		ASNs		IP prefix hijacked
73:2	128:2	5571	8190	212.2.0.0/19
254:2	498:3	16422	33770	41.223.58.0/24
254:3	498:4	16422	33770	41.223.59.0/24
254:4	498:5	16422	33770	41.223.57.0/24
268:2	498:6	17175	33770	41.220.224.0/24
268:3	498:7	17175	33770	41.220.225.0/24
268:4	498:8	17175	33770	41.220.226.0/24
268:5	498:10	17175	33770	41.220.229.0/24
268:6	498:12	17175	33770	196.201.228.0,22
328:2	489:2	19750	32004	207.181.144.0,24
342:2	421:16	20858	25184	80.75.13.0/24
351:2	466:3	21396	29606	194.1.150.0,24
351:3	466:4	21396	29606	91.199.151.0,24
351:4	466:5	21396	29606	195.177.192.0,23
383:7	564:2	23694	38513	202.87.191.0/24

TABLE 5. The incidents captured by the algorithm in Quarter 82

Quarter 82 without repetition events				
ASLoc:IPPLoc		ASNs		IP prefix hijacked
142:2	255:186	9229	17557	202.5.150.0/24
255:189	500:2	17557	36561	208.65.153.0/24

TABLE 6. The incidents captured by the algorithm in Quarter 83

Quarter 83 (last time of the hijack)				
ASLoc:IPPLoc		ASNs		IP prefix hijacked
339:2	1128:2	10143	38330	203.83.4.0/22
447:2	1027:2	13902	33694	208.71.120.0/21
549:188	1089:3	17557	36561	208.65.153.0/24
705:2	890:2	21792	27169	69.22.144.0/24
799:2	822:3	24213	24538	122.200.52.0/24

As it is noticeable from table 5, the algorithm identifies a duplicate

announcement when both AS17557 and AS36561 announce the same IP prefix (208.65.153.0/24) of the AS36561 in the 83rd quarter; from a detection perspective, this is equivalent to a potential hijacking incident, hence it is successful in detecting the event. One interesting feature of the analysis is that, when applying the YouTube dataset on the proposed detection method, beyond the expected result (the YouTube hijacking), the algorithm highlighted further multiple announcement events relating to other IP prefixes. The following section provides an overview of these identified events.

A. New Incidents

This section analyses IP prefix incidents as reported by the proposed detection method using the dataset during Pakistan Telecommunication hijack incident. The purpose of choosing a specific incident is to determine the strength and potential limitations of the algorithm. The detection outcomes of the algorithm are categorised into three classes: a) same organisation with same IP prefix (low probability hijack), b) different organisation with same IP prefix (high probability hijack) and c) no exit to an AS with one organisation. These incidents are summarised into the following points:

a) Same Organisation with Same IP prefix:

DoD Network Information Centre (DNIC), Comcast Cable Communications Holdings and 24/7 Real Media, in the US [12]. MDNX Enterprise Services and MDNX Internet Limited, in the UK [13]. Indonesia Network Information Centre, PT Arsen Kusuma and Digital Satellite PT, in Indonesia [13].

b) *Different Organisations with Same IP prefix:*

Cable Communications Inc with DH Data Centres Inc, Criteo Corp with Business Information Group, Townsend Analytics LTD with Viztek, Flagler Hospital Inc with Trident Systems Inc, in the US [12]. BHARTI Airtel Ltd with Karuturi Telecom Pvt Ltd, NetConnex Broadband with Borwood UK Network, in the UK [13]. Pakistan Telecommunication Company Limited (Pakistan) with Speed Cast Limited in (Hong Kong) [13]. Pakistan Telecom (Pakistan) and YouTube (US) [12][13]. Exetel Pty Ltd and Speedweb Network, Australia [13].

c) *Non-existent to the AS with one Organisation:*

New Skies Satellites Inc (US) with an anonymous AS [12][13]. Afranet Tehran (Iran) with an anonymous AS [14].

In summary, the detection method found about 16 different IP prefix impersonations during the two hours timeframe of Pakistan Telecom hijacking. The first five events in a) are considered suspicious and the next nine cases in b) are certain hijacks and the last two events in c) are ambiguous because their ASes do not exist in IANA (Internet Assigned Numbers Authority).

A. Algorithm Challenges and Proposed Solutions

The first challenge is that when an edge router impersonates an IP prefix of an AS and the real owner does not announce this IP prefix in the same 15-minute chunk, in this case the algorithm will not be able to detect the hijack. The proposed 15-minute timeslot length could be increased to improve accuracy, but that would have a negative impact on granularity and

potential responsiveness. For example, if the period of the processed data is 2 hours (the total period of the hijacking), the algorithm could probably detect the IP prefix from the first compared chunk, assuming that the 74th quarter is the starting point of the exchangeable UPDATE messages.

If the period of collecting processed data is increased, the hijack (short-lived) might take place and finish before the algorithm starts comparing data (origin ASes against their IP prefixes) and finds the hijack. As there is no clear technique to determine the appropriate period of analysing data, it would be very difficult to find the hijack rapidly. Another suggested solution is, after comparing IP prefixes within one quarter, the same quarter can be compared to all the quarters in the day (cross-validation), but due to the amount of data exchange that would affect the performance of the detection method speed.

Another significant challenge is that an AS could impersonate a subspace/sub-prefix of a specific IP prefix. For example, YouTube CIDR (Classless Internet Domain Routing) is 208.65.152.0/22; any impersonator could announce from 208.65.152.0/22 up to 208.65.155.0/32. In other words, if Pakistan Telecom announces any of 208.65.152.0/22-32 - 208.65.155.0/22-32, instead of 208.65.153.0/24, it would still impersonate the ownership of the YouTube's IP prefixes. Moreover, there are two main factors, which could play important role in the accuracy of the detection method: routes aggregation and ASes confederation or reflection. In future, these two factors will be investigated in terms of their effect on the IP prefixes and ASes.

Periodically, some normal changes occur to the inter-domain routing (the internet) such as modifying organisation's ASNs and assigning closed organisations' IP prefixes to a new organisation during the algorithm detecting the IP prefix hijack. These factors could make the algorithm inaccurate because the history of collected data changes continuously. This issue needs to be investigated to reduce false positives. The described challenges encounter the algorithm running on a single router; if the system is extended to allow the exchanging of information and work collaboratively with different copies of the algorithm on remote routers, its accuracy and strength would significantly increase. A possible collaboration architecture is introduced in the following section.

V. Proposed detection architecture

This section describes a possible architecture that allows the proposed method to aggregate data collaboratively on several different routers. The aim of the architecture is to allow BGP routers to jointly benefit from the independently identified events on each router and, subsequently, lead to higher accuracy when detecting anomalous behaviour.

A. Architecture Method and the Advantages

IP prefix hijacks might not influence the impersonated AS greatly until they spread out over many different ASes, therefore the algorithm has to work collaboratively to allow it to prevent the spread out of invalid routes and then to limit and reduce the impact of the hijack. The detection algorithm is connected to the BGP independently and categorises network events, but may benefit from

sharing and receiving data from other similar routers in order to detect the effect of the attack rapidly. The BGP updates are collected and aggregated by a router over a specific operational timeslot (e.g. between 10 minutes and 2 hours) as anomaly detection becomes stale with higher aggregation slots. In case of detecting a suspicious route, an alarm is sent to the all neighbours reporting the invalid route. Routers that run the algorithm should work together in order to improve the reliability and timeliness of the information derived from the UPDATE messages.

The algorithm should run over each router, connected to the internet, with different random checked chunks. In addition to the use of BSA, making the routers work collaboratively and independently would make detection very fast and would not require any modifications of the infrastructure of the BGP routers. Figure 8 shows the general structure of the detection method when it is linked to the BGP routers.

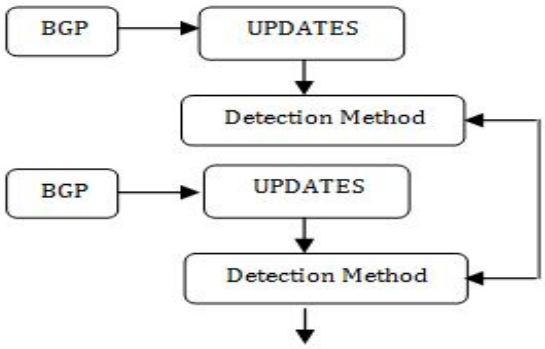


Fig. 8. Method architecture of detecting and preventing the hijack

Moreover, if some routers try to omit the hijack intentionally (e.g. do not have the system), the other routers identify and publicise the anomaly. Each BGP router has to have a chance to suppress any suspicious routes to prevent itself from spreading hijacked routes out.

B. The Effectiveness of the Architecture over the Algorithm

The advantage of a collaborative architecture in the BGP context is that each router can only check its own received update packets so there is no load to the algorithm to find out the hijack. Another advantage of this collaboration is that the check will be periodic and timeslot variable. For example, if Router A cannot detect the hijack at 1:15 AM because it is not the time slot to do the check, there must be another copy of the algorithm in the neighbouring routers doing the check and detecting the hijack faster.

VI. Conclusion and future work

This paper proposed an IP prefix hijack detection method, using 24th of Feb 2008 incident as a case study to build a trustworthy algorithm in order to detect new hijacks. However, a number of other national (regional) and international (out-region) BGP announcement anomalies have been detected during the aforementioned period. From the results, it is clear that the algorithm can work accurately but could also omit some events. For instance, if an AS announces an IP prefix in the absence of the real origin AS, the algorithm will not be able to detect the impersonation when it works independently (non-collaboratively). We observed from the proposed architectural collaborative work how this issue could be addressed. In terms of router connections, some routers do not have a direct connection to the hijacker. In other words, the detection method is considered decentralised in order to collect direct information regarding the hijacker and detects the hijack faster. Another advantage of the decentralisation is that detection of anomalies can be done for various, partially overlapping timeslots. Another challenge of the algorithm is that the hijacker could impersonate one of the

net-range IP prefixes (sub-prefixes) without the algorithm observation. As a consequence, the algorithm needs to be able to check the IP prefixes of an AS. In future, the proposed approach may provide further insight and refining of the rationale behind organisations announcing the same IP prefix with different ASN. This is needed in order to distinguish between normal BGP operations and malicious ones and address the false positive errors. The algorithm can detect the hijacks with 69% and fail with 31%. These false positives will be investigated based on two suspicious factors, which might make the algorithm not very accurate. First factor is directly related to the BGP policy such as route aggregation and ASes confederation while the second factor regards to the management of providing ASNs and IP prefixes such as allowing to some organisation to have more than one AS or IP prefix.

REFERENCES

[1] Y. Rekhter, "A Border Gateway Protocol 4 (BGP-4) Status," p. 67, 1995.

[2] I. O. de Urbina Cazenave, E. Kosluk, and M. C. Ganiz, "An anomaly detection framework for BGP," 2011 Int. Symp. Innov. Intell. Syst. Appl., pp. 107–111, Jun. 2011.

[3] M. Wählisch and T. C. Schmidt, "Towards Detecting BGP Route Hijacking using the RPKI," no. ii, pp. 103–104, 2012.

[4] H. Cao, M. Wang, X. Wang, and P. Zhu, "A Packet-Based Anomaly Detection Model for Inter-domain Routing," 2009 IEEE Int. Conf. Networking, Archit. Storage, pp. 192–195, Jul. 2009.

[5] D. Meyer, "Index of bgpdata," University of Oregon, 2003. [Online]. Available:

http://archive.routeviews.org/bgpdata/. [Accessed: 11-Nov-2013].

[6] H. Balakrishnan," How YouTube was Hijacked ," no. May. 2009.

[7] C. D. Marsan, "Six worst Internet routing attacks." [Online]. Available: http://www.networkworld.com/news/2009/011509-bgp-attacks.html. [Accessed: 28-Jan-2014].

[8] Ripe, "Index of source of bgpdump." [Online]. Available: http://www.ris.ripe.net/source/bgpdump/libbgpdump-1.4.99.11.tar.gz. [Accessed: 06-Jul-2014].

[9] MathWork, "Cell Arrays - MATLAB & Simulink - MathWorks United Kingdom." [Online]. Available: http://uk.mathworks.com/help/matlab/cell-arrays.html. [Accessed: 12-Jul-2014].

[10] A. Dalal, "Searching and Sorting Algorithms," no. 100. pp. 1–13, 2004.

[11] D. Meyer, "University of Oregon Route Views Project," 2003. [Online]. Available: http://www.routeviews.org/. [Accessed: 12-Jun-2014].

[12] ARIN (American Registry for Internet Numbers), "Whois-RWS." [Online]. Available: http://whois.arin.net/ui/query.do. [Accessed: 01-Feb-2014].

[13] A. (Asia P. N. I. Centre), "APNIC - Query the APNIC Whois Database." [Online]. Available: http://wq.apnic.net/apnic-bin/whois.pl.

[14] RIPE, "RIPE Database Query." [Online]. Available: https://apps.db.ripe.net/search/query.html. [Accessed: 13-Mar-2014].

Improving IP prefix hijacking detection by tracing hijack fingerprints and verifying them through RIR databases

Hussain Alshamrani
Centre for Security, Communications and Network Research (CSCAN) Plymouth University
Plymouth, UK
hussain.alshamrani@plymouth.ac.uk

Bogdan Ghita
Centre for Security, Communications and Network Research (CSCAN) Plymouth University
Plymouth, UK
bogdan.ghita@plymouth.ac.uk

Abstract—In spite of significant on-going research, the Border Gateway Protocol (BGP) still encompasses conceptual vulnerability issues regarding impersonating the ownership of IP prefixes for ASes (Autonomous Systems). In this context, a number of research studies focused on securing BGP through historical-based and statistical-based behavioural models. This paper improves the earlier IP prefix hijack detection method presented in [1] by identifying false positives showing up due to the organisations that may use multiple ASNs (Autonomous System Numbers) to advertise their routes. To solve this issue, we link a Verification Database to the previously proposed detection method to improve the accuracy. The method extracts the organisation names (unique code) and associated ASNs from different ASN delegators and RIRs (Regional Internet Registries), more specifically the RIPE (Reseaux IP Europeans) dump database [2] in order to evaluate the method. Since the organisation name is not available in the BGP updates, the data are extracted and

processed to produce a structured database (Verification DB). The algorithm excludes false positive IP prefix hijack detection events in the SFL (Suspicious Findings List) introduced in [1]. Finally, the algorithm is validated using the 2008 YouTube Pakistan hijack event and the Con-Edison hijack (2006); the analysis demonstrates that the improved algorithm qualitatively increases the accuracy of detecting the IP prefix hijacks, specifically reducing the false positives.

Keywords—RIPE database; ASNs and IP prefix delegators; information correlation; false positives

I. Introduction

BGP remains the protocol of choice for core Internet interconnectivity. Although a number of BGP security issues have been identified for almost two decades, the protocol remains vulnerable to IP prefix attack [3]. These weaknesses cause serious attacks and open the door for attacker to perform spam attack [4], traffic interception and DDoS [5]. On Oct, 2014 Sharon Goldberg pointed out that the main reasons why BGP is taking so long to be secured is that, apart from the fact that the BGP security solutions are not deployable, BGP lacks a single centralised authority, each organisation deploys its own routing security solution autonomously, so a complete or mass deployment is unlikely to take place [3].

Previous studies tried to detect IP prefix hijacks based on monitoring routers' stability, but their methods could not reliably distinguish IP prefix hijacks from normal events, such as power cut-off and submarine cuts [6]. In addition, RPKI (Resource Publication Infrastructure) was put forward to detect BGP route hijacking, but the

system had several false positives and negatives and need further refinements [7].

Lastly, some methods propose analysing the routing tables in order to detect IP prefix hijacks, but they are likely to have a limited impact, as organisations may refuse to provide their routing tables or are unable to timely detect a hijack event [8]. In addition, previous solutions do not support collaboration among routers to detect the IP prefix hijacks, collaboration could limit the attack spreading out and affecting a large number of networks.

This paper aims to address the false positives caused by the limitations of the algorithm in [1]. After investigating the main reasons we found that route aggregation and AS confederation or reflection BGP operations do not affect the accuracy of the IP prefix detection algorithm proposed previously [1] although they have a direct effect on the routes.

One of the factors affecting the algorithm is that big organisation can announce their routes with multiple different ASNs; to counteract this issue, a novel combination of RIRs and ASNs delegation database and BGP updates [9] is proposed in order to accurately and timely detect IP prefix hijacking events.

In section II, the paper discusses the previous detection method and the limitations of its algorithm. Section III shows the creation of the Verification DB based on the RIPE database. In Section IV we describe the proposed improvements to the IP prefix detection method based on the information from the Verification DB, together with findings and algorithm challenges. Section V describes the collaboration between routers to detect the

IP prefix hijacks before it spreads out. Section VI proposes a general structure of the detection method to be linked with the BGP routers so it can work efficiently. The paper finishes with the conclusion and future work in section VII.

II. Previous detection method

The detection method presented in [1] consists of three main parts: pre-processing, analysis and the algorithm, as shown in Figure 1. This section shows that the algorithm did not have a mechanism to validate the output. It makes decision directly and displays the result either benign or malicious.

The next two subsections explain that by providing an overview of the algorithm functionality and highlight its limitations, specifically the shortcomings that we aim to improve in this paper. BGPdump is a tool used to convert updates from binary data to ASCII data.

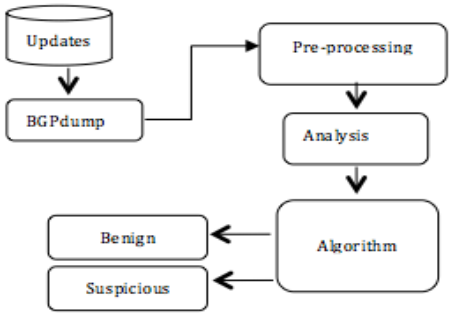


Fig1. Previous structure of the IP prefix hijack detection method

A. Algorithm

The algorithm has three objectives – firstly, associates the announcer (origin AS) with each advertised IP prefix. Secondly, removes duplications of associated origin ASes and IP prefixes. Finally, identifies any IP prefixes that were announced by more than one AS.

The algorithm receives the origin ASes and their IP prefixes from two different data

sources, every fifteen minutes. Origin ASes are mapped on their IP prefixes using a cell array in MatLab [1] allowing assignment of one ASN to multiple IP prefixes.

Data reduction is then applied to the dataset by removing duplicated origin ASNs and IP prefixes, which allows the algorithm to categorise faster the input dataset in order to detect suspicious announcements. After computing unique origin ASes and their associated IP prefixes, the algorithm compares the IP prefix of each AS to the IP prefixes of all origin ASes reported during each time interval to find out IP prefixes that were advertised by multiple ASes.

The analysis performs a comparison between individual AS-IP prefix rows in the cell array using the BSA (Binary Search Algorithm) [10] O(log N) due to its ability to execute array comparisons exponentially faster than linear search algorithm (LSA) [11]. The algorithm lists the detected incidents (suspicious hijacks) in a new matrix composed of two columns.

Table 1 shows the format of the Mapping Cell Array for Origin ASes and IP prefixes. The comparison part shows the outputs as normal or suspicious routes. However, the algorithm in [1] has some false positives as it is going to be explained in the following subsection.

TABLE.1 Example of Mapping Cell Array in quarter 82

Origin ASes	IP prefixes
137	369760021
151	369760021;369760023; 697600524
174	139438524; 244296124

B. Previous algorithm limitation

The algorithm proposed in [1] has a significant limitation, as it cannot take into account organisations using multiple different ASNs to advertise their own routes. To address this limitation, this paper introduces a Verification database to be

included in the detection method in order to enhance the accuracy of the algorithm. Since BGP updates lack the organisation names (codes), we extract data from the RIRs and process them to produce a dataset that links the AS numbers to the unique codes for the organisations that own them.

III. RIR-based mapping of AS numbers and organisations

This section discusses the processing of RIR information (specifically the RIPE Whois database [2]) to enhance the BGP update fields used as input and support the algorithm described in [1] to reduce the false positives.

A. Extracting and numerating organisations' ASNs and their unique codes from RIPE dump database

As part of the RIR registration, each organisation has a unique code to uniquely identify it. For instance, in RIPE, ORG-YE1-RIPE field represents Yahoo in Europe but ORG-HBp1-RIPE represents HSBC Bank plc. The Verification DB is processed in three phases.

1. PHASE 1

This phase extracts the ASNs and organisation codes fields from the RIPE dump database and stores data into corresponding fields, *aut-nums* and *orgs*, such as autonomous system number AS20535 and its code ORG-IG12-RIPE.

2. PHASE 2

Since RIPE includes ASNs without an associated organisation code (name), the incomplete records are filtered out, which does inherently limit the capabilities of the presented method because they confuse the order of

searching the ownership of specific IP prefix and mix them up.

3. PHASE 3

The organisation code (name) field is structured as an array and was created to include all organisations codes that facing every ASN in RIPE. Each organisation code (organisation name) is divided into three parts (ORG, IG12 and RIPE for example) and saved in an array called ORG. Second and third index in ORG array respectively represent the organisation name and data resource (e.g. RIPE). Currently, the most important part is the second field of the array because it uniquely identifies the organisations.

The third field of the organisation code array represents the database (e.g. RIRs or ASN delegator) that provided the record; this helps to differentiate between multiple database source owners. To optimise the analysis, these two parts are converted to numeric data. Table 2 shows one record of the final structure format of the Verification DB. First column is used as a primary key to be linked to ASNs in the Suspicious Findings List [1].

TABLE.2 Example of the final format to the Verification DB

ASNs	ORG codes and sources
200912	18191226

B. Filtering organisations with one ASN

Given the method focuses on organisations with more than one ASN in order to refine the results, all organisations that have only one ASN are filtered out, allowing the algorithm based on [1] to parse a significantly smaller dataset in order to

determine whether suspicious IP prefix hijacks are real or not.

In the case of RIPE database from February 2015 [9], the size of the Verification Database before filtering out organisations with only one ASN was 25580 records, reduced to 6283 records through filtering. The improved detection method verifies its results (suspicious hijacks) based on the reduced Verification DB. The general structure of processing the Verification Database is shown in Figure 2.

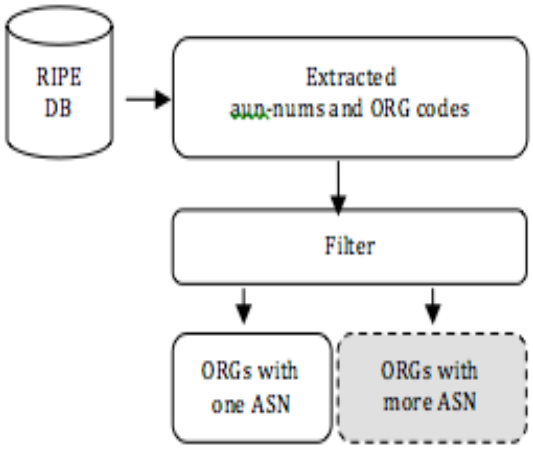


Fig. 2 Structure of Verification DB of organisations that have multiple ASNs

IV. Improved detection method

This section discusses the use of the Verification DB in the IP hijack detection method [1]. The encompassing algorithm validates its outputs based on this database. It also demonstrates the results of the detection method after the improvement.

In the previous work [1], the algorithm directly translates the results into two categories, normal and suspicious, but it does not verify the decision against organisations owning multiple ASes. To expand, if an organisation relocates a prefix between two of the ASes it owns, the algorithm would flag the change as a suspicious event; in fact, given both ASes are owned by the same organisation, it is likely that it is due to addressing and logistics ASes and IP prefixes management rather than a hijacking incident. In this paper, we introduce the Verification DB to check against the owners of the ASes involved in the suspicious events.

The Verification DB maps the autonomous system numbers and the corresponding organisations owning them. The extended comparison allows us to verify if a suspicious event is due to an IP prefix being migrated between ASes owned by distinct organisations. If an IP prefix is indeed migrated between ASes owned by different organisations, the event is further flagged as suspicious; if however the migration is between ASes of the same owner, the algorithm concludes that the change is not a suspicious event and continues with the search.

Figure 3 provides a block-diagram overview of the improved detection method, including input from RIRs into the decision process. In the diagram, the Extensional Block provides the required functionality for the RIR information and verification DB processing.

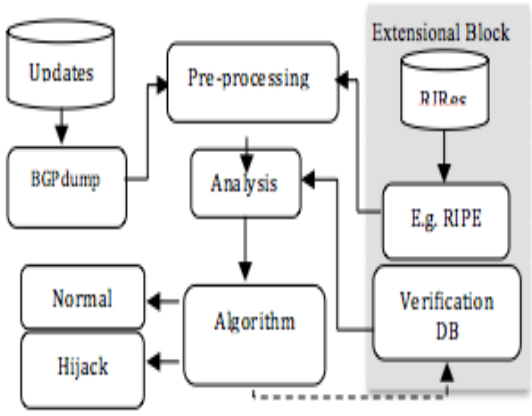


Fig. 3 Improved Structure of the IP prefix hijack detection method [1]

V. Validation of improved detection method

The algorithm proposed in the previous section was applied to two incidents: the whole day of Pakistan and YouTube hijacking day (24/02/2008) and the day of the Con-Edison hijack (22/01/2006). In other words, before the algorithm takes a decision with the suspicious routes, it checks out if two suspicious routers that were impersonating the same IP prefix exist in the Verification DB with one organisation name; if so, they are ignored otherwise the

advertisement will be flagged as a hijack. Pseudo code below explains the steps of the validation in details. The algorithm develops the accuracy of the suspicious results that were already caught by searching for the signature attack of IP prefix hijackings. It takes each two suspicious ASes in the list of Suspicious Finding List and searches for them in VerificationDB, which contains organisations that have more than one ASN; if they exist in the SFL, the ASes will be removed from suspicious list as they are not a real signature for the IP prefix hijackings.

```
Suspicious = dlmread (Suspicious_Finding_List);
suspiciouslen = length (suspicious);
VerifDBLen = length(ORGsWithMultiASN);
CASE = 1;
ORGCODE = [1 0; 2 1];
WHILE CASE <= suspiciouslen
   ASN1 = suspicious (CASE, 4);
   ASN2 = suspicious (CASE+1,4);
   CHECK = 1;
   WHILE CHECK < VerifDBLen
      ASN3=ORGsWithMultiASN (CHECK, 1);
      IF (ASN1 == ASN3 OR ASN2 == ASN3)
         IF (ASN1 == ASN3)
           ORGCODE (1,1)=CASE;
           ORGCODE (1,2)= . . .
           ORGsWithMultiASN (CHECK, 2);
         ELSEIF (ASN2 == ASN3)
           ORGCODE (2,1)= CASE;
           ORGCODE (2,2)= . . .
           ORGsWithMultiASN (CHECK, 2);
         END
      END
      IF (ORGCODE (1,2) == ORGCODE (2,2)
          ORGCODE (1,1) == ORGCODE (2,1))
             suspicious(CASE-1: CASE+1,:)=[];
          suspiciouslen= . . .
          length(suspicious);
        ORGCODE (1,2)=0;
        ORGCODE (2,2)=1;
      END
      CHECK= CHECK+1;
   END
   CASE= CASE+3;
END
```

A. Findings

The improved algorithm added functionality has two advantages: it can detect multiple occurrences of the same incident and allows the algorithm to identify organisations that announce their routes with more than one ASN. In the specific example of the YouTube hijack, the algorithm from [1] identifies 1767 incidents; following the analysis of repeated incidents, 975 unique incidents can be identified. Parsing the analysis through the Verification DB, the number of Suspicious hijacks drops to 969, due to the SLF suspicious hijack exclusions. Following a similar processing, the events from (the 22nd Jan 2006) do not show any

improvement because the incidents took place outside RIPE, so the Suspicious Findings List from RIPE is empty (none of suspicious results in the findings list is in the RIPE database). Thus, the algorithm needs several sources such as AFRINIC (Africa Region), APNIC (Asia/Pacific Region) and ARIN (North America Region) to improve its accuracy.

B. Algorithm challenges and solutions

Since the Verification DB uses only the RIPE database as a case study, the results would still include false positives but with lower percentage. The algorithm would be more accurate if the number of different sources (RIRs' and ASN delegators' database) used for the Verification DB increases. This challenge can be addressed as described at the end of the previous sub-section.

Second challenge is that the RIRs and ASNs delegators' databases need to be updated regularly and concurrently with the changes to ASNs and organisation names. Third difficulty is the algorithm detects IP prefix hijacks based on off-line analysis.

Furthermore, some organisations do not include their code in their associated record in the RIPE database. In addition, some RIRs do not keep historical records of old Whois registrations details. Once a record is updated or deleted, the old record is not stored in an archived database. As a result, the algorithm cannot evaluate organisation names and ASNs changes when it compares past suspicious hijacks to the current Verification DB.

Finally, prefix hijacks may be transparent for the algorithm on a subset of routers due to partially propagated prefix updates; therefore routers need to work

collaboratively to compare and aggregate update information with their neighbours. The following section discusses the steps of this collaboration.

VI. Proposed detection architecture

This section describes a possible architecture that allows aggregating data collaboratively on several different routers. The aim of the architecture is to allow BGP routers to jointly benefit from the independently identified events on each router and, subsequently, lead to higher accuracy when detecting anomalous behaviour.

A. Architecture method and the advantages

Routers that run the hijack detection algorithm should work together in order to improve the reliability and timeliness of the information derived from the UPDATE messages. An IP prefix hijack might not significantly affect traffic exchanged with the impersonated AS until it spreads to multiple/different ASes; to alleviate the effect of the hijack, the algorithm has to work collaboratively to prevent the propagation of invalid routes. The detection algorithm operates independently from BGP and categorises network events, but may benefit from sharing and receiving data from other neighbouring routers in order to detect the effect of the attack rapidly. The BGP updates may be collected and aggregated by a router over a specific operational timeslot, while bearing in mind that anomaly detection becomes stale with higher aggregation slots. In case of detecting a suspicious route, an alarm of the invalid route would be sent to all neighbours.

The algorithm should run in each router, based on the different information received. In addition to the use BSA (of Binary search algorithm), making the routers work

collaboratively and independently would increase the detection speed and would not require any modifications of the infrastructure of the BGP routers. Figure 4 shows the general structure of the improved detection method when linked to the BGP routers.

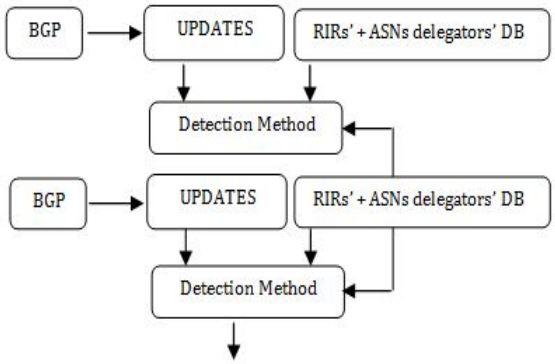


Fig. 4 Improved detection method architecture of detecting and preventing the spread out of hijacks

Moreover, if some routers do not actively run the detection system, the other routers may identify and publicise the anomaly. By doing so, each BGP router will have a chance to suppress any suspicious routes to prevent itself from further propagating the hijacked routes.

B. The effectiveness of the architecture over the algorithm

The advantage of a collaborative architecture in the BGP context is that each router can only check its own received update packets so there is no load to the algorithm to find out the hijack. Another advantage of this collaboration is that the check will be periodic, with timeslot starting times distributed over time. For example, if Router A cannot detect the hijack at 1:15 AM because it is not the time slot to do the check, there may be another copy of the algorithm in the neighbouring routers doing the check and detecting the hijack faster.

VII. Conclusion and future work

A new framework was proposed to enhance the accuracy of a previously proposed method for IP prefix hijack detection. The framework extracts the

unique code and associated ASNs of organisations from different RIRs; the algorithm then excludes previously detected IP prefix hijacks that are likely to be false positives. After proposing the framework, its efficiency is validated on the Pakistan IP hijacking event from 24th Feb 2008 and the Con-Edison hijack (22nd Jan 2006). The analysis used the RIPE dump database from the two respective dates as a case study to evaluate the proposed framework. In the evaluation, the algorithm was able to improve the accuracy of the IP prefix hijacks, reducing the false positives by 0.61% (18 suspicious hijack) for the two events.

From the results, it is clear that the algorithm can work accurately but also could omit some events; more specifically, several incidents from 22nd Jan 2006 were still false positives, since the analysis was based only on the RIPE database. Additionally, if an AS announces an IP prefix in the absence of the real origin AS, the algorithm will not be able to detect the impersonation when it works independently (non-collaboratively).

In terms of router interconnectivity, some routers do not have a direct connection to the hijacker. In other words, the detection method ought to be decentralised in order to collect direct information regarding the hijacker and detect the hijack faster. Another advantage of the decentralisation is that detection of anomalies can be done for various, partially overlapping timeslots. Another challenge of the algorithm is that the hijacker could impersonate one of the net-range IP prefixes (sub-prefixes), event that may be transparent for the algorithm. Last, the period gap (synchronisation) between fetching BGP updates and the current status of the ASN of an organisation, together with the IP prefixes changes, could

have a negative impact on the accuracy of the algorithm.

In future, the proposed approach may provide further insight into and refine the rationale behind organisations announcing the same IP prefix with different ASN. This is needed in order to distinguish between normal BGP operations and malicious ones, and then address the false positive errors.

REFERENCES

[1] H. Alshamrani, B. Ghita, and D. Lancaster, "Detecting IP prefix hijacking using data reduction-based and Binary Search Algorithm," in *2015 Internet Technologies and Applications (ITA)*, 2015, pp. 78–84.

[2] John Stamatakis, "Pen Test Live: Download Database," 2014. [Online]. Available: http://www.pentestlive.com. [Accessed: 09-Jan-2014].

[3] S. Goldberg, "Why is it taking so long to secure internet routing?," *Commun. ACM*, vol. 57, no. 10, pp. 56–63, Sep. 2014.

[4] J. Schlamp, J. Gustafsson, M. Wählisch, T. C. Schmidt, and G. Carle, "The Abandoned Side of the Internet: Hijacking Internet Resources When Domain Names Expire," in *arXiv preprint arXiv: ...*, 16-Dec-2015, pp. 188–201.

[5] P. Vervier, O. Thonnard, and M. Dacier, "Mind Your Blocks: On the Stealthiness of Malicious BGP Hijacks," in *Proceedings 2015 Network and Distributed System Security Symposium*, 2015, no. February, pp. 8–11.

[6] I. O. de Urbina Cazenave, E. Kosluk,

and M. C. Ganiz, “An anomaly detection framework for BGP,” in *2011 International Symposium on Innovations in Intelligent Systems and Applications*, 2011, pp. 107–111.

[7] M. Wählisch, O. Maennel, and T. C. Schmidt, “Towards detecting BGP route hijacking using the RPKI,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, p. 103, Sep. 2012.

[8] H. Cao, M. Wang, X. Wang, and P. Zhu, “A Packet-Based Anomaly Detection Model for Inter-domain Routing,” in *2009 IEEE International Conference on Networking, Architecture, and Storage*, 2009, pp. 192–195.

[9] D. Meyer, “University of Oregon Route Views Archive Project,” *University of Oregon*. [Online]. Available: http://archive.routeviews.org/bgpdata/2008.02/UPDATES/. [Accessed: 05-Oct-2013].

[10] A. Dalal, “Searching and Sorting Algorithms,” 2004. [Online]. Available: http://www.cs.carleton.edu/faculty/adalal/teaching/f04/117/notes/searchSort.pdf. [Accessed: 08-Dec-2014].

[11] A. Horvath, “Quicksort, binary search and linear search performance - far from what you believe,” 2012. [Online]. Available: http://blog.teamleadnet.com/2012/02/quicksort-binary-search-and-linear.html. [Accessed: 05-Aug-2013].

IP Prefix hijack detection using BGP attack signatures and connectivity tracking

Hussain Alshamrani
Centre for Security, Communications and Network Research (CSCAN)
Plymouth University
Plymouth, UK
hussain.alshamrani@plymouth.ac.uk

Bogdan Ghita
Centre for Security, Communications and Network Research (CSCAN)
Plymouth University
Plymouth, UK
bogdan.ghita@plymouth.ac.uk

Abstract—In spite of significant on-going research, the Border Gateway Protocol (BGP) still suffers vulnerability issues specially regarding impersonating the ownership of IP prefixes of ASes (Autonomous Systems). In this context, a number of research studies focused on securing the BGP through historical-based and statistical-based behavioural models. This paper proposes a novel method aiming to detect IP prefix hijacking incidents based on tracking the behaviour of suspicious ASes. The detection method uses signature-based technique as a pre-process phase to separate suspicious announces (BGP updates) from benign announces. From a processing perspective, the outputs of signature-based algorithm are used as inputs for the detection method. Nine features will be extracted from the ASpath attributes of potentially suspicious ASes. The features are considered a combination of the behavioral characteristics of the routers in relation to their connectivity. Based on these features and the best five supervised learning classifiers, we identify the hijacks. Under different learning algorithms, the detection

method is able to detect the hijacks with a high accuracy especially with J48, which can detect the hijacks with 96%.

Keywords—BGP4; Machine learning; ASN; IP prefix hijack signatures, features; RIRs Whois databases

I. Introduction

BGP remains the protocol of choice for core Internet interconnectivity. Although a number of BGP security issues have been identified and partially addressed for almost two decades, the protocol remains vulnerable to IP prefix attacks. This weakness leads to significant stability issues for the network, and may be used as a vehicle for blackhole traffic attackers [1], spamming [2], DDoS, and man-in-the-middle attacks [3]. In addition, hijackers may exploit redirecting BGP traffic for hijacking cryptocurrecny transactions [4].

In a review of existing approaches, Goldberg indicated that the main reason BGP is taking so long to be secured is that, apart from its deployment challenges, the infrastructure lacks a central authority, as each organisation autonomously deploys its own solution, so a complete or mass deployment is unlikely to take place [5]. A traditional method employed by prior research to detect IP prefix hijacks based on anomaly detection and monitoring the stability of the encompassing routers. Nonetheless, such methods could not reliably distinguish IP prefix hijacks from normal events, such as power cut-off or submarine cable cuts [6]. Lastly, some detection methods analyse routing tables (table-based) in order to detect IP prefix hijacks, but organisations may refuse to provide their routing tables [7]. Vervier et al. noted that methods based on monitoring anomalies to detect IP prefix hijacks are still suffering from high false positive rates

[3]. They also pointed out that prevention methods BGP IP prefix hijack are still facing large-scale and deployment issues [3]. Due to several reasons, such as performance issues on large routing systems or impracticability of approaches like S-BGP [8], the threats still exist nowadays [9]. Wubbeling et al. pointed out security based on origin authentication and asymmetric encryption are not feasible nowadays, because it is not yet implemented in broadly used hardware and business processes of ASes [9]. In addition, the RPKI (Resource Publication Infrastructure) system is one of IP prefix hijacking detection systems put in place to prevent BGP route hijacking. However the system had several false positives and negatives and needs further refinements. The system is based on tracing the hierarchical relationships of the address space were given by IANA, RIRs and big ISPs to customers. The Route Origin Authorizations (ROAs) is cryptographically signed and published in repositories. Routers can download these repositories using trusted tool and then upload them into [10].

Zhang et al. pointed out the importance of signature-based and anomaly-based in modern intrusion detection together with their inherent drawbacks – uncertainty for signature-based methods and inability to detect new attacks for anomaly-based analysis [11]. Furthermore, connectivity model is a new approach used recently to trace the behaviour of opportunistic networks. Kathiravelu argues that a paradigm shift from mobility models to connectivity model [12]. As a result, we decided to simulate these models wok and build our method based on the connectivity behaviour changes of suspicious ASes to detect BGP IP prefix hijacks.

In this paper we propose a detection method that can trace the behaviour of suspicious ASes and detects the IP prefix hijacking. The detection method relies on the connectivity behaviour of suspicious ASes to their neighbours. Hijack signatures are checked to separate suspicious announces from benign announces. From this connectivity we can extract several parameters such as number of sender and receiver neighbours for suspicious ASes, the victim and the hijacker. In order to validate the accuracy of the method, a dataset of UPDATE messages was collected and used using the Route View project of University of Oregon, covering the 24hour period of the 24th of February 2008; the day was chosen as it is the day when Pakistan Telecom intended to restrict local access to YouTube from their citizens. However, it advertised an IP prefix owned by YouTube and blocked access to YouTube [13] for approximately two hours [14]. For the detection method validation purposes, a number of supervised machine-learning classifiers based on Split Test option were used and resulted in accuracy rates of up to 96%.

This paper is organised as follows: in section II we present the components of the IP prefix hijacking detection method. The section also crosschecks the BGP updates for detecting ASes' hijack footprint, while section III extracts some features based on the connectivity of suspicious ASes. Section IV discusses the methodology of the classification and testing the behaviour of suspicious ASes while V evaluates the accuracy of the detection method based on the results of learning algorithms. The conclusions and future work are outlined in the last section.

II. Detection method

In the last decade, Machine Learning started to be used to detect anomalies on the network traffic. We plan to use this subfield of computer science to detect IP prefix hijack in the BGP. The Machine Leaning has different learning approaches to mine data such as supervised learning, semi-supervised learning, unsupervised learning, reinforcement learning and deep learning. Because supervised-learning pre-learns malicious and benign instances, it is considered more accurate than other learning types; therefore the datasets of suspicious ASes will be structured in supervised format. The IP prefix hijack detection method is composed of four main components as it is shown in figure 1: IP prefix hijack Parser, Feature Extractor (FE), Labeller and different ML (Machine Learning) classifiers.

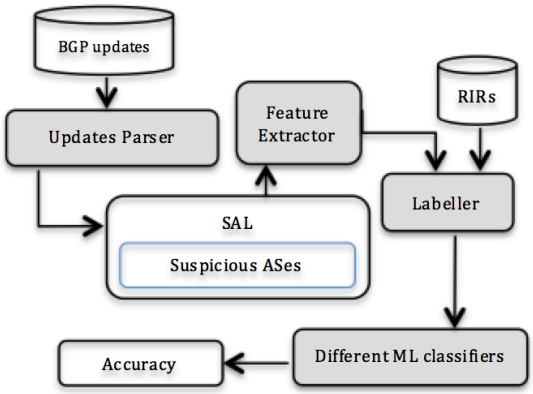


Fig 1. Detection method using signature-model-based combination

A. Parsing announcements

The detection method is based on BGP announce update messages downloaded from the Route View project created by University of Oregon. The parser splits update messages into equal timeslots and identifies multiple ASes that announce the same IP prefix. Further, data reduction is applied to reduce the search area of BGP update messages. The Parser maps every AS to all its unique IP prefixes announced in the period the announces determined to be checked in. Mapped ASes and their

prefixes are put in cell array. Table 1 shows the ASes when they are mapped to their IP prefixes before they are parsed.

TABLE 1. Mapped ASes with their prefixes

Unique ASes	Unique Prefixes
AS37	'198.91.71.0/24'
AS100	'63.115.54.0/24'
AS801	18.168.0.0/24
	158.173.176.0/20
AS1239	128.30.0.0/15
	18.168.0.0/24
	18.168.1.0/24
AS1299	158.173.176.0/20
AS14807	'63.115.54.0/24'
AS27064	'198.91.71.0/24'

Each record in the cell array, in table 1, is composed of pair data (suspicious unique AS, unique prefix/prefixes). Each record is compared to the whole list of the ASes and their prefixes to structure table 2. This table has two columns, first one represents different ASes claiming the same IP prefix and the second column represents the prefix is announced by two different suspicious ASes, all duplicated suspicious incidents are removed. The algorithm employs binary search algorithm to trace prefix hijacks very quick. Binary search algorithm executes in algorithmic time and with Big O. This table represents data are saved in the SAL (Suspicious ASes List), which displays the output format of suspicious ASes.

TABLE 2. Suspicious ASes

Different ASes		Same IP prefix
AS1239	AS801	18.168.0.0/24
AS801	AS1299	158.173.176.0/20
AS37	AS27064	198.91.71.0/24
AS100	AS14807	63.115.54.0/24

Parser passes its outputs to the Feature Extractor, which is responsible for extracting nine features from candidate suspicious ASes connectivity to their direct neighbours. These features will be discussed in detail in section III.

B. Labelling incidents

The behaviours of suspicious ASes received from the Feature Extractor are labelled based on the RIRs (Regional

Internet Registries) databases. Since AS numbers and prefixes are delegated by several organisations and the detection method only based on the RIRs for exploring the hijacker from the victim, there are some incidents cannot be labelled because some suspicious ASes or the impersonated prefixes are not available in the RIRs. The dataset in the table 2 is labelled into two classes either malicious or benign. The detection method needs to label the outputs of the Feature Extractor in order to specify the possible patterns of hijacking behaviors. Each nominated suspicious AS is investigated based on the five regional registries: AfriNIC, APNIC, ARIN, LACNIC and RIPE NCC. The strategy of labeling the events is based on three main rules:

- If both of suspicious ASes own the route, they are both marked with OWNER, and the event is considered benign.
- If one of suspicious ASes owns the prefix, it is marked with OWNER, while the other suspicious AS is marked as HIJACKER, and the event is considered malicious.
- If neither of the suspicious ASes owns the prefix, we tag them with NOTSURE, and then the event is labeled as AMBIGUOUS.
- If the suspicious ASes do not exist, they will be ignored.

TABLE 3. Suspicious ASes investigator dataset

AS 1	AS 2	AS1 STATUS	AS2 STATUS	LABEL
100	250	OWNER	OWNER	BENIGN
200	10	ATTACKER	OWNER	MALICIOUS
300	50	NOT SURE	NOT SURE	AMBIGUOUS

BENIGN represents semi-hijacks. Semi-hijacks means one organisation could own a block of ASes and announce a prefix with different ASNs (AS numbers). This operation gives the same signature of the real hijack but in reality it is benign. MALICIOUS represents the real hijacks. AMBIGUOUS events will be removed

from the dataset and we only keep records are labeled as BENIGN or MALICIOUS as it shown in table 3. Labelled instances are passed as inputs to the ML (Machine Learning) component to be trained and then classified. The detection method uses the best learning algorithms had been used on previous applications to evaluate the proposed features. These algorithms were selected based on the most used supervised-learning classifiers in the last recent years. In our work, we use percentage-split test option for training and testing the datasets.

III. Features extraction

The majority of previous anomaly detection methods [6] [15] [16] extract features based on the stability of routers. However, these methods always fail to detect the IP prefix hijack or are rather poor at differentiating it from other anomalies because features are extracted based on the stability of routers are not feasible. There are lots of issues that can make routers not stable such as blackouts outages, worms. All features are going to be extracted in this work are new and extracted in a different way. Our extraction method is based on the connective structure (topology) of suspicious ASes to their direct neighbors. The detection method extracts features from the direct location to the issue, which is ASPATH attribute, to differentiate the behavior of the hijackers and victims. ASPATH attribute is one of attributes the BGP has in order to deal with its policy. In order to see affected routers in the Internet infrastructure and calculate the connectivity between routers when an edge router impersonates the ownership of a prefix is owned by another edge router (AS), we use Network Analysis and Visualization (NAV) toolbox [17]. The

NAV toolbox can help to plot the topology of suspicious ASes and trace their connectivity to their neighbours. When the behaviors of the suspicious ASes are computed they will be stored in triple format, first two locations for suspicious ASes and third location for hijacked prefix, (AS1, AS2, Prefix).

A. Feature computation

The behavior of each suspicious AS will be calculated from its connectivity to the direct neighbors and the result of two the ASes are subtracted to give the relational behavior value of pair suspicious ASes. This relational behavior value of all features (instance) represents the pattern of the two ASes claiming the hijacked prefix. Since, the relational behavior between the two suspicious ASes can be negative, in some cases, we need to take the absolute value of the differences from equation 1. For example, we assume that we have four pair edge routers in two four different ASes, as in figure 2. Victim router is in AS1, Hijacker router in AS2, Owner router in AS3 and another Owner in AS4. Victim router and Hijacker represent real hijack whilst Owner routers are pair suspicious hijack. Victim receives some announcements but from two neighbors while Hijacker receives its announcements from one neighbor. This shows that upper pair routers in AS1 and AS2 have different number of sender neighbors. However, two pair suspicious routers in AS3 and AS4 have the same number of sender neighbors. . Owner in AS3 and 4 are considered suspicious because they carry the signature of the hijack; two different routers in two different ASes announce one prefix. The scenario of calculating the behavior of pair suspicious routers is applied to the remaining features. The results of each pair ASes are subtracted

and put in a column vector to represent the behavior of pair suspicious ASes This column vector represents the relational behavior of two suspicious incidents. If the relational behavior value is negative, equation 1 will be applied to remove the sign. S_{AS1} represents the behavior of first suspicious AS while S_{AS1} reflects the behavior of second suspicious AS and S_r represents the relational behavior between two ASes. SAS1 and SAS2 could be both owner or one is victim and another is hijacker.

$$S_r = |S_{AS1} - S_{AS2}| \quad (1)$$

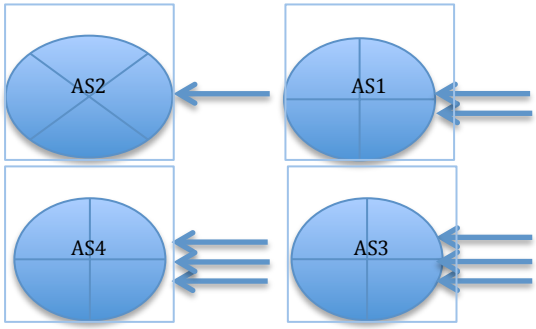


Figure 2. Routers connectivity and routes travel

TABLE 4. Features of suspicious ASes

NO	Type	Features
1.	Connectivity	# of repeated incidents
2.		# of receiver neighbors
3.		# of sender neighbors
4.		# of first propagators of suspicious routes
5.		# of shared receiver neighbors
6.		# of shared sender neighbors
7.		# of shared first propagators of suspicious routes
8.		# of connections between suspicious ASes
9.		Are they neighbors?

All features in table 4 are extracted from the ASPATH attribute. From the connectivity of suspicious ASes and their relational behavior we extract 9 features. Feature 1 is extracted based on the observation that says unintentional hijacks behavior, such as misconfiguration, do not impersonate more than one prefix whereas man-made prefix hijack often attack different ASes at same time. This feature has to distinguish between deliberate hijacks and unintentional hijacks. Features 2-7 are based on the connections of the routers to suspicious ASes. Specifically, features 2-4 focus on the direct neighbors

(routers) of suspicious ASes while features 5-7 analyze shared direct neighbors between suspicious ASes and features 8 and 9 identify direct and indirect connections between the suspicious ASes. These features should reveal the similar and different patterns of suspicious ASes behaviours.

B. Sampling data

Table 5 shows a snapshot of the instances calculated based on the proposed features appear in table 4. Based on the labeling method presented in section II, each instance is given either 0, if it is malicious or 1, if it is benign. Symbols F1-F9 represent the number of features while C indicates whether the event is a hijack or not. Each feature is stored in a separate column vector. These column vectors are concatenated with the class column vector to give a dataset composed of 10 columns, including observation classes, and 340 instances.

TABLE 5. Features after labeling

F1	F2	F3	F4	F5	F6	F7	F8	F9	C
2	22	1	665	0	1	0	0	0	0
2	0	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	1
1	0	0	66	0	0	0	0	0	0
2	7	0	7	1	0	0	1	1	1
1	5	0	0	0	0	0	0	0	1
10	0	1	0	0	0	0	0	0	1

We collected 680 suspicious ASes, which represents pair of suspicious ASes behavior in one record. Instead of putting each suspicious incident in one record we put pair of suspicious ASes in one record because we already separated benign announces from suspicious announces in parsing phase. In the next phase we need to distinguish pair benign ASes' behavior from one malicious and one benign ASes' behavior from each other. Basically, the number of suspicious ASes is 640 but because of the mechanism of sampling the

dataset, which based on the relational behavior of suspicious ASes instead of putting the patterns of the suspicious ASes in a separate record, this number is dropped down to 340.

After sampling suspicious ASes in 340 pair suspicious ASes records, we also ignore 133 records since the registration details of some suspicious ASes does not show by any of RIRs Whois. The reason is probably because the suspicious ASNs or the hijacked prefixes are delegated by another ISPs (Internet Service Providers). The new size of the dataset will be dropped to 207 instances.

Another important thing has to be taken into account is getting rid of redundant instances, which means all duplicated records will be removed from the dataset because there is no need for similar events but this also makes the dataset decreasing. Although the size of the dataset is decreasing, the performance of detecting hijacks is increased because we only collect incidents what we are sure from. With removing duplicated patterns of suspicious ASes behavior will make the detection method goes over a large number of BGP updates very fast during live communication between BGP routers. It would be good idea if RIRs or interested originations keep the history of the hijacker and victim ASes in addition to collecting BGP update messages. That will help interested researchers to have a large dataset to work on.

After labeling instances based on the RIRs Whois validation and removing repeated suspicious observations, the new size of the dataset will be 113 instances (suspicious ASes patterns) by 9 attributes (features). If proposed learning algorithms in the following section can differentiate

the patterns for malicious and benign observations, that means the extracted features are useful and built in a high efficient way. Next section will determine the quality of the features and evaluate the detection method.

IV. Classification and evaluation

This section discusses the method is used to divide the dataset that already received by Feature Extractor and labelled by the Labeller into training and test datasets. The method is called Split Test; it is a simple way to use one dataset to both train and estimate the performance of the features on unseen data. The incidences are classified based on the following steps:

- The dataset is split randomly into 80% training dataset and 20% test dataset for each learning algorithm.
- Picked classifiers start with building the models of different learning algorithms and test unseen instances of the suspicious ASes.
- Every classifier's parameters are adjusted repeatedly till we find the best parameters work efficiently with the features.
- The best result of each classifier is registered to be compared with the other good classifiers can work with the dataset.
- Based on the offset of the percentage of false positives and false negatives of the classifiers, the best result is picked.

A. Best classifiers with different applications

In 2008, Xindong Wu et al. had studded the best algorithms were used in data mining in the past decades. Based on the study it is found that the most influential data-mining algorithms are allocated in 10 top algorithms. The study was about the best learning algorithms among several methods such as classification, clustering, statistical learning, association analysis and link mining [18]. Table 6 shows the

best 10 supervised learning algorithms that can be used in classification in different applications.

TABLE 6. Top 10 algorithms in data mining [18]

Algorithm	Category	Learning Types	Families
C4.5 (J48)	Classification	Supervised	Decision tree
K-Means	Clustering	Unsupervised	Clusters
SVM	Statistical	Supervised	-
Apriori	Association	Unsupervised	Associations
EM	Clustering	Unsupervised	Clusters
PageRank	Link mining	Unsupervised	-
AdaBoost	Classification	Supervised	Ensemble
KNN	Classification	Supervised	Lazy
Naive Bayes	Classification	Supervised	Bayes
CART	Classification	Supervised	Decision tree

Another study has taken place for investigating the best learning classifiers in 2014. The study compared 179 classifiers for 17 families and over 121 different databases and found the best classifiers are Random Forest versions [19]. RandomForest belongs to rule-based family and categorised as supervised learning. In addition, In 2014 Kaur and Chhabra claimed that improved J48 used recently to increase the accuracy rate of classification [20].

B. Classifiers selection

We have two different deep studies of the best algorithms in data mining; first study [18] was based on the research community and how the best algorithms are used widely in different area in data mining while the second study [19] was empirical study performed by some experts in data mining. Both of studies are important because they cover each other limitations. The studies covered two types of learning but we only interested in classification supervised-learning algorithms as we can have a prior picture of the percentage of benign and malicious data and know them before they are given to the algorithms. Our dataset was

prepared to suit the specifications of different algorithms. It is numeric, dose not has missing values, its attribute values are natural, discrete and its class is binary which makes the classification to classifier easier. According to the strongest two attribute evaluators, PCA (Principal Component Analysis) finds out that the dataset have only one redundant attribute and the remaining 8 attributes are relevant while SVM (Support Vector Machine) attribute evaluator is considered all attributes are useful.

Based on the features of the algorithms such as accuracy, speed and offset of having false positive and negatives and the ability to deal with the structure of the dataset the classifiers are going to be used chosen. The detection method is going to use J48, which is considered the improved version of C4.5 and C5.0 since it has several advantages and can work with the structure of our dataset. Generally, decision tree is fast at classifying unknown instances and easy to interpret for small-sized tree (dataset). In addition, decision tree can handle discrete attributes, work well in the presence of redundant attributes and robust to the effect of outliers therefore two classifiers, RandomForest and CART (SimpleCart), are going to be tried too. Furthermore, additional two supervised-learning classifiers will be used because of their solid makeup such as k-NN (k-Nearest Neighbour) and NB (Naïve Bayes). Both NB and K-NN support complex decision function or non-linear decision boundary to isolate multidimensional data and different classes

C. Testing The Detection Method

Proposed algorithms are randomly fed with different percentage of training

datasets. However the classifiers accuracy rises up when the training dataset is 80% as in table 7. According to the changing of algorithms parameters the accuracy of the classification is registered. All algorithms parameters values need to be changed continuously to suit the feature structure. However, SimpleCart (CART) classifier gives 95% under its default values of the parameters. Table 7 shows that J48 is the best algorithm can work with the detection method. However, the detection method put the classifiers error rate into account to pick the best algorithm. The computation of the error rate will be worked out in the next subsection.

TABLE 7. Results based on Rule and Tree machine learning algorithms

Family	Algorithm	Training dataset	Test dataset	Accuracy
		Percentage-split	Percentage-split	
Trees	J48	80 %	20%	96%
Lazy	KNN			91%
Bayes	NB			87%
Trees	CART			95%
Trees	RF			91%

D. False Negatives Calculation

A and B in table 8 represent correctly and incorrectly classified instances for both classes, benign and malicious. 0 represents malicious class and 1 reflects the class of benign instances. It is notable from table 8 that algorithms have more difficulty to classify benign observations than malicious observations except in Naïve Bays and RandomForest but generally all algorithms work well with the features. J48 and CART classify the dataset and come up with 0 incorrectly classified malicious observations and 18 correctly classified malicious observations while k-Nearest Neighbours and has 0 Incorrectly classified malicious observations and 10 correctly classified observations. However, Naïve Bayes and RandomForest have 2 incorrectly

classified malicious observations and 16 correctly classified instances.

In terms of benign observations classification, RandomForest is considered the best classifier among five algorithms because its incorrectly classification is 0. J48, NB and CART have the same accuracy rate of detecting benign data while KNN is considered the worst as it has 2 instances classified incorrectly.

If the total number of choosing the classified observations is considered, we find that k-Nearest Neighbours is only classifying 15 instances picked out of 113 observations but the remaining algorithms are all equal and classifying 23 instances are chosen randomly from 113 observations. Based on these notes and the offset of false positives and false negatives, the best algorithm is elected.

TABLE 8. Confusion matrix testing for best classifiers

Algorithm	Train dataset	A	B	Classified as
J48	80 %	18	0	A=0
		1	4	B=1
KNN	80 %	10	0	A=0
		2	3	B=1
NB	80 %	16	2	A=0
		1	4	B=1
CART	80 %	18	0	A=0
		1	5	B=1
RF	80 %	16	2	A=0
		0	5	B=1

From the two following equations 2 and 3, we can compute the percentage error of the false positives and false negatives for each algorithm, where MF_n represents malicious false negative while BF_n represents benign false negative. Every algorithm selects a number of instances randomly. This number represents the classified dataset size.

$$MF_n = \frac{incorectly\ clissified\ instances}{classifed\ dataset\ size} \quad (2)$$

$$BF_n = \frac{incorectly\ clissified\ instances}{classifed\ dataset\ size} \quad (3)$$

For each algorithm the false negatives and false positives of two classes are calculated from equation 2 and 3. If we take the average of the result, it will give the percentage error of the whole algorithms to distinguish benign and malicious instances, which means the accuracy of the detection method according to the whole five algorithms. The calculation comes up with 0.03 false negative for malicious class and 0.05 false positives for benign class, as it is shown in table 9. The percentage error of detecting malicious patterns is less than the detection of benign patterns.

TABLE 9. Error of detecting malicious and benign hijack signature

Algorithms	Malicious False negatives	Benign False negatives
J48 (80%)	0	0.04
KNN (80%)	0	0.13
NB (80%)	0.08	0.04
CART (80%)	0	0.04
RF (80%)	0.08	0
AVG	0.03	0.05

Figure 3 visualises the percentage error of classifying real hijacks and semi-hijacks that labelled in section II-b. It shows that the false negative is less than the false positive in three classifiers (J48, KNN and CART) and explores the best algorithm based on the trade-offs of the false positives and negatives of the classifiers. From the trade-off perspective, the graph shows that J48 and CART are the best two algorithms.

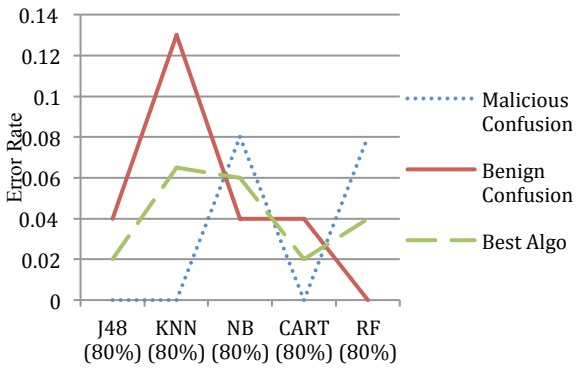


Fig 3. Algorithms tried with the detection method

Summarising, the detection method works in a good efficiency with the whole classifiers in figure 3. However, the best

two classifiers can work with extracted features are J48 and CART. Our experiment supports Kaur and Chhabra study, which held in 2014 and says J48 increases the accuracy rate of the classification. However, as the J48, KNN and CART are good with detecting real hijacks and RF is considered the best of detecting semi-hijacks, the detection method would work better if we can combine RF with J48, KNN or CART. In other words, It is assumed that the detection method works better, if RF can be combined with one of the three classifiers has zero false positive during classifying benign observations.

V. Conclusion

In conclusion, this paper discussed a novel method to detect IP prefix hijacks in the BGP. The method uses the BGP updates as input and parses the AS origin of announcements. A Feature Extractor receives caught suspicious ASes an input and extract features based on the behavior of suspicious ASes connectivity. The feature values of the suspicious ASes are given to a Labeller to be labeled with two classes, benign and malicious. The labeller uses the information of RIRs registration details of the organizations to know the victim and the hijacker and to determine if the suspicious caught incident is benign or malicious. The outputs of the Labeller, which is the dataset, will be given to different supervised classifiers. We create our own accurate dataset by checking the ownership of ASes and IP prefixes via RIRs and omit ambiguous and not exit suspicious ASes. Different learning algorithms were used to choose the best classifier works with the features. Generally, the result of the method is encouraging and very good as the percentage of false positive and false

negative is less that 10% and the accuracy of the best classifier (J48) is 96%.

RERENCES

[1] H. Ballani, P. Francis, and X. Zhang, “A study of prefix hijacking and interception in the internet,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, p. 265, Oct. 2007.

[2] P.-A. Vervier, Q. Jacquemart, J. Schlamp, O. Thonnard, G. Carle, G. Urvoy-Keller, E. Biersack, and M. Dacier, “Malicious BGP hijacks: Appearances can be deceiving,” in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 884–889.

[3] P. Vervier, O. Thonnard, and M. Dacier, “Mind Your Blocks: On the Stealthiness of Malicious BGP Hijacks,” in *Proceedings 2015 Network and Distributed System Security Symposium*, 2015, no. February, pp. 8–11.

[4] G. Valadon and N. Vivet, “Detecting BGP hijacks in 2014 BGP Hijacking for Cryptocurrency Profit Reported by Dell SecureWorks on August 7 2014,” 2014. [Online]. Available: http://www.nosuchcon.org/talks/2014/D3_04_Guillaume_Valadon_Nicolas_Vivet_detecting_BGP_hijacks.pdf.

[5] S. Goldberg, “Why is it taking so long to secure internet routing?,” *Commun. ACM*, vol. 57, no. 10, pp. 56–63, Sep. 2014.

[6] I. O. de Urbina Cazenave, E. Kosluk, and M. C. Ganiz, “An anomaly detection framework for BGP,” in *2011 International Symposium on Innovations in Intelligent Systems and Applications*, 2011, pp. 107–111.

[7] H. Cao, M. Wang, X. Wang, and P. Zhu, “A Packet-Based Anomaly Detection Model for Inter-domain Routing,” in *2009 IEEE International Conference on Networking, Architecture, and Storage*, 2009, pp. 192–195.

[8] S. Kent, C. Lynn, and K. Seo, “Secure Border Gateway Protocol (S-BGP),” *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 582–592, 2000.

[9] M. Wubbeling, T. Elsner, and M. Meier, “Inter-AS routing anomalies: Improved detection and classification,” in *2014 6th International Conference On Cyber Conflict (CyCon 2014)*, 2014, pp. 223–238.

[10] M. Wählisch, O. Maennel, and T. C. Schmidt, “Towards detecting BGP route hijacking using the RPKI,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, p. 103, Sep. 2012.

[11] K. Zhang, A. Yen, X. Zhao, D. Massey, S. Felix Wu, and L. Zhang, “On Detection of Anomalous Routing Dynamics in BGP,” in *Proceedings of the International IFIP-TC6 Networking Conference 2004*, vol. 5, 2004, pp. 259–270.

[12] T. Kathiravelu, A. Pears, and N. Ranasinghe, “Connectivity Models : A New Approach to Modeling Contacts in Opportunistic Networks,” *Proc. Eighth Int. Inf. Technol. Conf. 2006*, p. 185, 2006.

[13] H. Balakrishnan, “How YouTube was Hijacked,” 2009. [Online]. Available: http://web.mit.edu/6.02/www/s2012/handouts/youtube-pt.pdf.

[14] C. D. Marsan, “Six worst Internet routing attacks.” [Online]. Available:

http://www.networkworld.com/news/2009/011509-bgp-attacks.html. [Accessed: 28-Jan-2014].

[15] J. D. Gardiner, "Multiple Markov Models for Detecting Internet Anomalies from BGP Data," in *2009 DoD High Performance Computing Modernization Program Users Group Conference*, 2009, pp. 374–377.

[16] N. Al-Rousan, S. Haeri, and L. Trajkovic, "Feature selection for classification of BGP anomalies using Bayesian models," in *2012 International Conference on Machine Learning and Cybernetics*, 2012, pp. 140–147.

[17] MathWorks, "Network Analysis and Visualisation," 2015. [Online]. Available: http://uk.mathworks.com/help/bioinfo/network-analysis-and-visualization.html. [Accessed: 10-Nov-2015].

[18] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, Jan. 2008.

[19] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?," *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, 2014.

[20] G. Kaur and A. Chhabra, "Improved J48 Classification Algorithm for the Prediction of Diabetes," *Int. J. Comput. Appl.*, vol. 98, no. 22, pp. 13–17, 2014.

IP Prefix Hijack detection using BGP connectivity monitoring

Hussain Alshamrani
Centre for security, Communications and Network Research (CSCAN) Plymouth University
Plymouth, UK
hussain.alshamrani@plymouth.ac.uk

Bogdan Ghita
Centre for security, Communications and Network Research (CSCAN) Plymouth University
Plymouth, UK
bogdan.ghita@plymouth.ac.uk

Abstract—In spite of significant on-going research, the Border gateway protocol (BGP) still encompasses conceptual vulnerability issues regarding impersonating the ownership of IP prefixes for ASes (Autonomous Systems). In this context, a number of research studies focused on securing BGP through historical-based and statistical-based behavioural models. This paper suggests a novel method based on tracking the connectivity of suspicious ASes, which are received from a program tracing IP prefix hijacking signature. The paper uses Full Cross-Validation test to investigate the accuracy of the invented method and studies the similarity and differences between malicious and benign observations before they are classified. Classification might not be the appropriate technique to deal with IP prefix hijack detection on its own; therefore we propose to combine the two methods (signature and classification-based) in order to cover the limitations of both techniques. From a processing perspective, the outputs from signature-based method are used as inputs for the classification-based. The main features are extracted from the ASpath attributes of potentially suspicious ASes.

The features are considered a mixture of the behavioural characteristics of connectivity among routers. The best five supervised classifiers were used in the previous researches and go with the characteristics of dataset will be used in this paper to evaluate the detection method. Under different learning algorithms, Random Forest and J48 classifiers, the detection method is able to detect the hijacks with 81% accuracy.

Keywords—BGP4; Machine learning; ASN; IP prefix hijack; features; RIRs Whois databases, route, MOAS, routes

I. Introduction

BGP remains the protocol of choice for core Internet interconnectivity. Although a number of BGP security issues have been identified for almost two decades, the protocol remains vulnerable to IP prefix attacks. This weakness leads to significant stability issues for the network, and may be used as a vehicle for black-hole traffic attackers [1], spamming [2], DDoS, and man-in-the-middle attacks [3]. In addition, hijackers may exploit redirecting BGP traffic for hijacking cryptocurrency transactions [4]. On April 2015 Schlamp pointed out to the reason that leads to hijacking of routes. For example, the main reason threatens the BGP security is emerging from abandoned Internet resources such as address blocks or AS numbers. In other words, when the DNS names expire, the attacker reregister domain names which are referenced by corresponding RIR (Regional Internet Registries) database objects [5]. 20% of the whole IPv4 address space is presently allocated but not above-board announced; this unused space is the ideal environment for such malicious BGP hijack events [3]. To solve this issue, our methods require

organisations to announce their IP prefix at least once in order to advertise their ownership to the IP prefix block.

In a review of existing approaches, Goldberg indicated that the main reason BGP is taking so long to be secured is that, apart from its deployment challenges, the infrastructure lacks a central authority, as each organisation autonomously deploys its own solution, so a complete or mass deployment is unlikely to take place [6].

A traditional method employed by prior research has been to detect IP prefix hijacks based on anomaly detection and monitoring the stability of the encompassing routers. Nonetheless, such methods could not reliably distinguish IP prefix hijacks from normal events, such as power cut-off or submarine cable cuts [7]. Lastly, some detection methods analyse routing tables (table-based) in order to detect IP prefix hijacks, but organisations may refuse to provide their routing tables [8]. Vervier et al. noted that methods based on monitoring anomalies to detect IP prefix hijacks are still suffering from high false positive rates [3].

They also pointed out that prevention BGP hijack methods are still facing large-scale and deployment issues [3]. Due to several reasons, such as performance issues on large routing systems or impracticability of approaches like S-BGP [9], the threats still exist nowadays [10]. Wubbeling et al. pointed out security based on origin authentication and asymmetric encryption are not feasible nowadays, because it is not yet implemented in broadly used hardware and business processes of ASes [10]. In addition, RPKI (Resource Publication Infrastructure) system is one of IP prefix hijacking detection systems put in place to

prevent BGP route hijacking. However the system had several false positives and negatives and needs further refinements. The system is based on tracing the hierarchical relationships of the address space were given by IANA, RIRs and big ISPs to customers. The Route Origin Authorizations (ROAs) is cryptographically signed and published in repositories. Every router has to upload the information [11].

As a case study, UPDATE messages were collected from the 24th of February 2008, using the Route View project of University of Oregon, when Pakistan Telecom intended to restrict local access to YouTube, but the advertised UPDATE messages blocked access to YouTube [12] for approximately two hours [13].

In this paper we implement a program to search for suspicious ASes and pass the result to another program to trace the behaviour of routers through their connectivity. From the behaviour we can extract several parameters such as direct and indirect neighbours, number of sender and receiver neighbours for both the victim and hijacker. These two programs form the structure of the detection method, which is a combination of signature and connectivity-based. Zhang et al. pointed out the importance of signature-based and anomaly-based in modern intrusion detection together with their inherent drawbacks – uncertainty for signature-based methods and inability to detect new attacks for anomaly-based analysis [14]. Furthermore, connectivity model is a new approach used recently to trace the behaviour of opportunistic networks. Kathiravelu argues that a paradigm shift from mobility models connectivity model [15]. As a result, we decided to combine signature-based and anomaly-detection-

based techniques to avoid their limitations when they work separately.

For the detection method validation purposes, we are going to use a number of supervised machine learning classifiers based on full cross-validation test technique. The highest accuracy of the hijack detection was achieved using J48 and RandomForest classifier where the accuracy reached 81%.

This paper is organised as follows: in section II we present the detection method of the IP prefix hijack. In section III we crosscheck the RIR Whois database with the outputs of validator to label incidents while in section IV we extract features based on the connectivity behaviour of suspicious routers. In section V we explore the similarity between suspicious and malicious observations before they are classified. Section VI discusses the methodology of the classification and testing the behaviour of suspicious ASes while VII evaluates the accuracy of the detection method based on the results of learning algorithms. The conclusions and future work are outlined in section VIII.

II. Detection method

In this section, we talk about how to connect the detection method of new parts to the previous work. The detection method is going to add a novel features are proposed to use supervised machine learning algorithms to detect IP prefix hijacking. Thus, we need a supportive part to do labelling for data. Tracer and validator blocks are beyond the scope of this paper.

Machine leaning has different learning approaches to mine data such as supervised learning, semi-supervised

learning, unsupervised learning, reinforcement learning and deep learning. However, the supervised-learning approach is more accurate and appropriate to the issue of impersonating others' IP prefixes issue; therefore, the dataset will be structured in supervised format.

The IP prefix hijack detection method is composed of five main parts as it is shown in figure 1: IP prefix hijack signature tracer, suspicious ASes validator, Labeller, Dataset Extractor and Organiser (DEO) and ML. However, this paper concern to only three blocks: the Model, ML and labeller part. Figure 1 shows the general structure of the detection method.

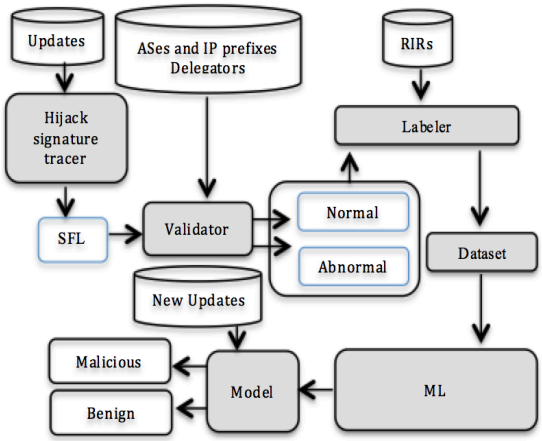


Fig 1. Detection method using combination of signature-based and connectivity-based

Tracer is signature-based algorithm receives update messages for specific period of time (15-minutes) and check them based on the IP prefix hijack signature. The algorithm uses two useful techniques data reduction and binary search algorithm to reduce search area of BGP messages. Table 1 shows the suspicious outputs the tracer caught. This table represents data were saved in the SFL (Suspicious Findings List), which exposes the output format of detected abnormal and suspicious routes.

TABLE 1. Suspicious finding list

Announcers	Neighbours	Routes
AS3	AS1239	128.30.0.0/15
		18.168.0.0/24
		18.168.1.0/24
AS3292	AS1299	158.173.176.0/20
	AS3549	
	AS8001	

Validator receives suspicious ASes as inputs and verifies them based on the database generated from RIRs Whois [16] and ASNs (Autonomous System Numbers) and IP prefixes delegators. We do that because BGP updates do not support organisation name data and the same signature of the hijack is showing up in the normal behaviours of routers, this conflict is called MOAS (Multiple Origin Autonomous System).

In case of the ASes and IP addresses ownership are not updated regularly in RIRs and their delegators, Labeller receives the inputs from validator and labels the outputs of the validator because the detection method is based on supervised learning approach. Since the RIRs operators do not save their old subscribers records, finding out the ownership history of some nominated suspicious routers make it very difficult to label some ASes. This method helps to decide and collect behaviour only from known ASes and ensure from and separate the benign and malicious ASes.

DEO is responsible for extracting anomaly detection features, organising data and classifying the behaviour of nominated benign and suspicious ASes. The DEO has 9 features extracted based on the suspicious routers connectivity. It categorises suspicious routes into two classes either normal or abnormal. The outputs of the DEO are passed as inputs to the ML (Machine Learning) block.

In ML (Machine Learning) we use five learning algorithms to evaluate the proposed features. In this part we use full-cross validation test option for training and testing dataset. The ML will give the final result accuracy of the detection method and use the detection model for detection new hijacks.

III. Labelling incidents

Sine RIRs (Regional Internet Registries) do not keep records of old Whois registrations details, this section intends to label the outputs of the validator in order to specify the ASes we are going to trace their behaviour during the hijacking history and then structure a very high accurate supervised learning dataset. Labeller still uses RIRs to build the dataset but it needs to filter confusing events that appear in the up to date Whois RIRs databases. Based on that, some nominated ASes were received from hijack signature tracer will be excluded from the outputs of the Validator as their ownership to the victim routes are ambiguous. Table 2 describes validator outputs before they are labelled.

TABLE 2. Validator outputs before labelling

AS1	AS2	IP prefix
3	27930	'190.14.196.0/24'
3	27930	'190.14.197.0/24'
37	27064	'198.91.71.0/24'
100	14807	'63.115.54.0/24'
100	14807	'65.204.11.0/24'
209	7018	'24.32.114.0/24'
209	2711	'64.53.21.0/24'
209	2711	'64.53.40.0/22'
209	6395	'66.212.81.0/24'

Each nominated suspicious AS is investigated based on the five regional registries: AfriNIC, APNIC, ARIN, LACNIC and RIPE NCC. The strategy of labelling the events is based on three main aspects:

- If both of suspicious ASes own the route, we mark them with OWNER, and then the event is benign.

- If one of suspicious ASes owns the route, it marks with ONER and HIJACKER, and then the event is malicious.

- If none of suspicious AS origins owns the route, we tag them with NOTSURE, and then the event is not labelled.

TABLE 3. Suspicious ASes investigator dataset

AS 1	AS 2	AS1 STATUS	AS2 STATUS	LABEL
100	250	OWNER	OWNER	BENIGN
200	10	ATTACKER	OWNER	MALICIOUS
300	50	NOT SURE	NOT SURE	AMBIGUOUS

AMBIGUOS events will be removed from the dataset and we only keep records were labelled as BENIGN or MALICIOUS as it shown in table 3. After extracting features as it is going to be in next section, each feature pattern will be given the class of its ASes event label.

IV. Features extraction

In order to see the pollution of the internet when an edge router impersonates the ownership of a route is possessed by another router, and the connectivity between suspicious routers during the hijacking, we use Network Analysis and Visualization [17] to plot the topology of suspicious ASes. Based on the behaviour of suspicious ASes we extract 9 features from their connectivity. The behaviour of each suspicious AS can be calculated separately. However, we interested in the event of two suspicious ASes impersonating same IP prefix; therefore we need to take the absolute value of the differences between calculated suspicious ASes behaviours from equation 1. For example, finding the number of receiving neighbours is calculated in two separate column vectors, one for AS1 and another for AS2, and we need to apply the equation 1 in order to put them in one vector column. This vector column

represents the behaviour of both ASes whether the event is malicious or benign. S_{AS1} and S_{AS2} indicate two sates either benign with benign or benign with malicious.

$$S_r = |S_{AS1} - S_{AS2}| \quad (1)$$

All features in table 4 were extracted from the behaviour of suspicious ASes (edge routers) are hidden in the ASPATH attribute. We briefly explain the purposes of these features. Since the innocent hijack does not occur for multiple different ASes, we extract the number of repeated incident in order to detect unintentional hijacks such as hijacks that occurred due to misconfiguration.

TABLE 4. Features of suspicious ASes

NO	Features
1.	# of repeated incidents
2.	# of receiver neighbours
3.	# of sender neighbours
4.	# of first propagators of suspicious routes
5.	# of shared receiver neighbours
6.	# of shared sender neighbours
7.	#of shared first propagators of suspicious routes
8.	# of connections between suspicious ASes
9.	Are they neighbours?

Generally, features 2-7 are based on the neighbourhood connectivity of suspicious ASes. Specifically, Features 2-4 concern about the direct neighbours of suspicious ASes while features 5-7 interests with shared direct neighbours between suspicious ASes. Feature eight and nine focus on direct and indirect connections between the suspicious ASes themselves. These features should reveal the similar and different patterns of suspicious ASes behaviours.

Table 5 shows a sample of the values of proposed features with their classes to detect the IP prefix hijacks. Each instance is labelled either with 0 if it is suspicious or 1 if it is benign. The type of pattern is

represented by the whole of the features. F1-F9 represents features and C represents the two categorical classes of the behaviour. In terms of feature organization and calculation, each feature is saved in a separate column vector after being calculating based on the connectivity of suspicious edge routers. These column vectors are concatenated to give a dataset composed of 10 columns, including classes, and 340 examples. Since the registration details of some suspicious ASes are not recorded and are not given in any of RIRs, we omit about 133 instances from the main dataset including malicious and benign samples. The dataset is built based on the rule explained in section III. The new size of the dataset will be dropped to have only 207 instances.

TABLE 5. Features after labelling

F1	F2	F3	F4	F5	F6	F7	F8	F9	C
2	22	1	665	0	1	0	0	0	0
7	0	1	0	0	0	0	0	0	0
12	0	1	0	0	0	0	0	0	1

Another important rule has to be taken into account is that getting rid of redundant instances, which means all repeated hijacks will be removed from the dataset because there is no need to similar events it. We observe that the size of the dataset is decreasing but with an increase in the accuracy of the data we are working on and getting rid of the redundancy. After labelling instances based on the RIRs Whois and removing repeated suspicious observation rules, the new size of the dataset will be limited to 113 instances. If the learning algorithms can distinguish the patterns of malicious and benign observations that mean the detection method was built in a high efficient way. Based on the results of the classifications we will evaluate the method.

V. Calculate data similarity

In this section we calculate the percentage of similarity and differences among benign and malicious observations. We invented our own algorithm to compute the similarity and differences of benign and malicious route patterns, which based on the XOR logical operator concept; the output is true if inputs are not alike otherwise the output is false.

Malicious and benign patterns are previously saved in one matrix. Malicious row observations are compared bit-by-bit against every benign sequence. X_b represents benign matrix row vectors and Y_m represents malicious matrix row vectors in formula 1 and 2.

$$X_b = [f1, f2, f3, f4, f5, f6, f7, f8, f9] \quad (2)$$

$$Y_m = [f1, f2, f3, f4, f5, f6, f7, f8, f9] \quad (3)$$

Based on these two vectors formulas we compare the behaviour of benign and malicious observations. The output of this comparison is stored in a matrix with either zero or one, zeroes represent similarity and ones represent differences. We calculate every benign pattern and by the end we come up with several matrices for one benign vector, the number of matrices is as same as the size of malicious dataset.

Formula 4 show the general computation of similarity and difference of each benign pattern to all malicious patterns, where for is the loop starts from the first observation in the benign dataset and ends at the size of it. X_{b_i} is benign observations and Y_m is the malicious observations. $\sum 0$ is the summation of similar cases and $\sum 1$ is the summation of different cases.

$$_{i=1}^{n} for\{X_{b_i}\ xor\ Y_m\}_{(\sum 1)}^{(\sum 0)} \quad (4)$$

We calculate similarities and differences means of benign and malicious patterns from below two formulas 5 and 6, where n is the number of number of similarities and differences, S_i represents the similarities and D_i represents the differences of every benign observation to all malicious observations in the dataset. $\bar{S}$ gives the mean of similarity for all benign observations and $\overline{D}$ returns with the mean of differences of all benign observations. Both similarity and differences patterns of malicious and benign patterns are calculated to only ensure that the quality of data has been calculated correctly. In other words, one operation either calculating the behavioural similarity or difference between benign and malicious is enough to show the quality of data. Symmetric graph of similarity and differences shows that the calculation of one operation is carried out properly as it shown in figure 3. Based on observation of the calculation we either use similarity or difference calculation for studying the quality of the dataset. This dataset has malicious and benign announcement patterns.

$$\bar{S} = \frac{\sum_{i=1}^{n} S_i}{n} \quad (5)$$

$$\overline{D} = \frac{\sum_{i=1}^{n} D_i}{n} \quad (6)$$

From the graph in figure 3 we realise that the range of differences of malicious and benign observations is limited between 4.9 and 8.9. Correspondingly, the similarity among malicious and benign observations is limited between 2.9 and 6.1. If the value of both calculations is subtracted, the result will equal 4, which represents the range of similarities and differences. This value is probably is not very big but enough to differentiate between malicious and benign behaviours.

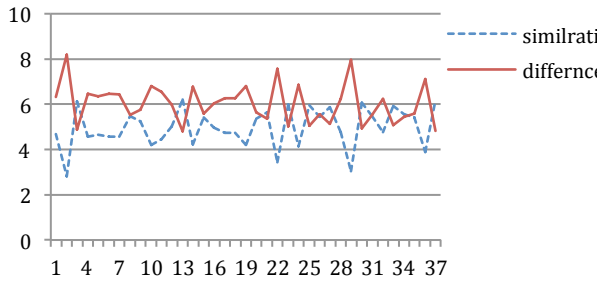


Fig 3. Similarity and differences between benign and malicious observations

Formula 7 computes the similarity percentage between malicious and benign behaviour, where total observations is equal to malicious observations plus benign observations. NS is the number of similarity was found in the whole dataset of benign and malicious observations. TO is the total observation of the dataset, which is 113. According to the formula, the percentage of the similarity is 0.07.

$$Similarity \text{ percentage } = \frac{NS}{TO} \quad (7)$$

The more similarity behaviours are greater than the difference behaviours, the more confusion could happen to learning classifiers. Since the similarity between malicious and benign datasets is good, we can use different classifiers in section IX to differentiate two patterns.

VI. Classification

In this section we discuss the method is going to be used to apply machine learning with cross-validation test to detect the IP prefix hijacks. After building the dataset, which is based on the connectivity of the routers, we are going to classify suspicious ASes based on the following steps:

a) The detection method firstly determines the appropriate method of Cross-Validation test is going to be used.

b) Since data is few we use Full Cross-Validation with all proposed learning algorithms as in figure 2.

c) Each algorithm repeats the classification with different parameters for many times in order

to observe the efficiency of the features and then the classifiers.

d) The best result of each classifier is saved to be compared to other classifiers' results.

e) Based on the offset of the number of false positives and false negatives, the best result among tried classifiers is determined.

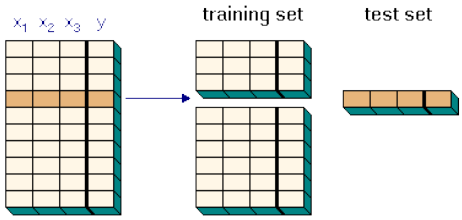


Fig 2. Full cross-validation technique

In 2008, Xindong Wu et al. had studded the best algorithms are using in data mining. Based on the research community it is found that the most influential data-mining algorithms are allocated in 10 top algorithms [18]. The study was about general types of learning algorithms such as classification, clustering, statistical learning, association analysis and link mining. However, we only interested in supervised-learning algorithms as we can have a prior picture of the percentage of benign and malicious data before they are given to the algorithms. On the other hand, another study has taken place for investigating the best learning classifiers in 2014. The study compared 179 classifiers for 17 families and over 121 different databases and found the best classifiers are Random Forest versions. RandomForest belongs to rule-based family and is considered supervised learning [19]. In addition, In 2014 Kaur and Chhabra claimed that improved J48 used recently to increase the accuracy rate of classification [20]. We have different deep studies of the best algorithms in data mining; first study was based on the research community and which the best algorithms were used widely in different area in data mining while the second study was empirical study performed by some experts in data mining. Both of studies are important

because they cover each other limitations. Based on these studies we are going to test the detection method using five supervised learning algorithms: J48 which is the improved version of C4.5 and C5.0, k-NN (k-Nearest Neighbour), NB (Naïve Bayes), CART and RF (Random Forest); and based on the features of the algorithms such as accuracy, fastness and offset of false positive and negatives, the classifier is chosen. We also can observe that the most of the best classifiers belong to supervised not unsupervised learning. Although Adaboost is a supervised classifier one of the best learning algorithms, it is going to be excluded because of its dependability on other classifier. Adaboost strength is acquired from other classifiers, which means the algorithm gives the same accuracy result of the classifier it based on therefore it will be ignored.

A. Testing

Proposed algorithms use full cross-validation technique, which also called leave one out cross-validation. In full cross-validation, we choose the largest fold (113), which is the size of the dataset, in order to enlarge training dataset and minimize the size of the testing dataset, as the original dataset is not big. Every single instance will be used as a test set and remaining data as training dataset. This idea helps to avoid the possibility that folds (testing datasets) have one or more instances have not been trained in other folds (training datasets).

For instance, suppose we have 100 instances and we use 10 cross-validation, the dataset will be divided into 10 chunks because 100 divided by 10 is equal 10; so each one has 10 instances but probably the tested route malicious behaviour in the

same fold of testing dataset. That means we might omit some hijacks if we do not maximize training dataset and minimize the test data set as much as we can. The smallest testing dataset and the largest training dataset we have, the more accurate evaluation of the detection method we receive. According to the changing of algorithms parameters the accuracy of the classification is registered as it shown in table 6. All algorithms parameters need to be adopted to suit the aim of the extracted features.

TABLE 6. Results based on Rule and Tree machine learning algorithms

Algorithm	Training dataset	Accuracy
J48	Full cross-validation	81%
KNN		79%
NB		76%
CART		81%
RF		81%

A. Error False Positive and Negatives Calculation

Confusion matrix in table 7 shows the accuracy of the detection method for both classes, malicious and benign. A and B represent correctly and incorrectly instances. Zero is the class of malicious instances and one is the class of benign instances. It is notable that the algorithms have difficulty to classify benign observations more than classifying malicious observations in the whole algorithms with slightly better in k-Nearest Neighbours and Random forest.

For instance, for malicious classification, J48 classified data and came up with 3 incorrectly classified malicious observations and 73 correctly classified observations. The case repeats itself in k-Nearest Neighbours, Naïve Bayes and Classification and Regression Tree algorithms. However, Random Forest has 7 incorrectly classified malicious instances and 69 correctly classified malicious

instances. In terms of benign classification, Random Forest is considered the best algorithm of detecting benign instances because it detected 23 benign instances correctly among 37 unique cases and the worst one is Naïve Bayes. Based on these notes and the offset of false positives and false negatives, the best algorithm will be elected.

TABLE 7. Confusion matrix testing for best classifiers

Algorithm	Train dataset	A	B	Classified as
J48	Full cross-validation	73	3	A=0
		18	19	B=1
KNN		73	3	A=0
		21	16	B=1
NB		73	3	A=0
		24	13	B=1
CART		73	3	A=0
		19	18	B=1
RF		69	7	A=0
		14	23	B=1

From two following equations, 8 and 9, we can compute the percentage error of the false positives and false negatives for the whole algorithms. NC_p Stands for Abnormal Confusion Percentage while AC_P represents Normal Confusion percentage.

$$NC_P = \frac{incorectly\ classified\ instnces}{correctly\ classified\ instnces} \quad (8)$$

$$AC_p = \frac{incorectly\ classified\ instnces}{correctly\ classified\ instnces} \quad (9)$$

If we take the percentage of false negatives and false positives for each algorithm and then the average of the whole algorithms, we come up 0.05 false negative and 1.15 false positives. Figure 3 shows that the false negative is less than the false positive in total but that does not explore the best algorithm; therefore, we judge the best algorithm based on taking the less false negative among malicious confusion computations and the less false positive in benign confusion.

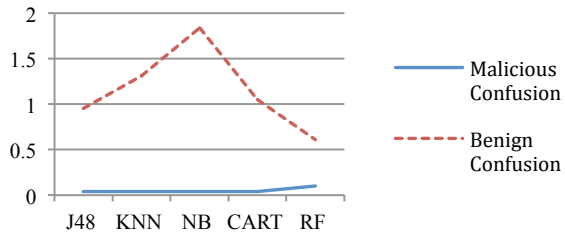


Fig 3. The best algorithm findings

From that rule we find the J48 is the best algorithm in terms of having less false negatives, if we take false positives of equal algorithms in to account. On the other hand, Random Forest is considered the best algorithm of detecting benign observations. If we take the offset of false positive and false negative of all classifiers results, Random Forest would be the best algorithm works with the features. However, the detection method would work better if we can combine these two algorithms to avoid learning implications for both algorithms. Formula 10 and 11 can compute the highest accuracy of the detection method when J48 and RF are combined, where HAM_m represents the highest accuracy of detecting malicious observations and HAM_b for detecting benign observations while $ICMO_{J48}$ and $ICBO_{RF}$ represent the incorrectly classified malicious and benign observations.

$$HAM_m = \left|\frac{ICMO_{J48} - ICMO_{RF}}{datset\ size}\right| \quad (10)$$

$$HAM_b = \left|\frac{ICBO_{J48} - ICBO_{RF}}{datset\ size}\right| \quad (11)$$

VII. Results and evaluation

Initially, the detection method was consisted of 12 features; these features are mixed of stability and connectivity observations of suspicious ASes were caught in tracing hijack signature phase. However, features were extracted based on the stability of edge routers are deleted as they make the detection very bad. As a result, the total number of features becomes 9. BGP packets are going to be

classified and evaluated based on the remaining 9 features. Generally, the five classifiers are suggested to be used can work with the extracted features in a high efficiency although all of them have false positives and false negatives but in low percentage.

The detection method result supports the studies that have been investigated in 2014, and said that the Random Forest versions and J48 are the best algorithms among classifiers [19] [20]. The false negative if we use J48 is 0.02 while the false positive is 0.15. On the other hand, if we use Random Forest as the classifier of the detection method, the false negative will be 0.06 and false positive 0.12, which means J48 is better than Random Forest since the number of false negative in J48 is less than the number of false negative in RF. If we want the detection method to be in the highest efficiency, it needs to work with J48 and RF integrally. For example, based on equation 10 and 11, false negative will be 0.04 and 0.03 false positive accuracy, which means its accuracy will be increased from 81% to 93%.

VIII. Conclusion and future work

In conclusion, this paper discussed a novel method to detect IP prefix hijacks in BGP. The method uses the extracted behaviour of suspicious ASes as inputs to the connectivity-based method, which in turn classify new BGP updates. Based on the suspicious ASes detected by the IP prefix hijacks Tracer data are classified into two classes, benign and malicious. Usually, researchers concern about the amount of data in their datasets. However, we interest in the uniqueness of suspicious and abnormal patterns, therefore the amount of data was few. Another reason

for making dataset small is that the algorithm excludes obvious normal observations from the dataset is going to be used for tracing routers connectivity. Moreover, there is no tool to give labelled accurate data of the historical incidents. As a result, we created our own accurate dataset by checking the ownership of suspicious ASes and IP prefixes through RIRs. Full Cross-Validation test method solves the issue of the size of the dataset because it is small. Five different learning algorithms and the best classifier works with the extracted features are picked. The result of the detection method is encouraging and very good as the percentage of false positive and false negative is less than 20% and the detection accuracy of the IP prefix hijacks is 81%.

REFRENCES

[1] H. Ballani, P. Francis, and X. Zhang, “A study of prefix hijacking and interception in the internet,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, p. 265, Oct. 2007.

[2] P.-A. Vervier, Q. Jacquemart, J. Schlamp, O. Thonnard, G. Carle, G. Urvoy-Keller, E. Biersack, and M. Dacier, “Malicious BGP hijacks: Appearances can be deceiving,” in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 884–889.

[3] P. Vervier, O. Thonnard, and M. Dacier, “Mind Your Blocks: On the Stealthiness of Malicious BGP Hijacks,” in *Proceedings 2015 Network and Distributed System Security Symposium*, 2015, no. February, pp. 8–11.

[4] G. Valadon and N. Vivet, “Detecting BGP hijacks in 2014 BGP

Hijacking for Cryptocurrency Profit Reported by Dell SecureWorks on August 7 2014," 2014. [Online]. Available: http://www.nosuchcon.org/talks/2014/D3_04_Guillaume_Valadon_Nicolas_Vivet_detecting_BGP_hijacks.pdf.

[5] J. Schlamp, J. Gustafsson, M. Wählisch, T. C. Schmidt, and G. Carle, "The Abandoned Side of the Internet: Hijacking Internet Resources When Domain Names Expire," in *arXiv preprint arXiv: ...*, 2015, pp. 188–201.

[6] S. Goldberg, "Why is it taking so long to secure internet routing?," *Commun. ACM*, vol. 57, no. 10, pp. 56–63, Sep. 2014.

[7] I. O. de Urbina Cazenave, E. Kosluk, and M. C. Ganiz, "An anomaly detection framework for BGP," in *2011 International Symposium on Innovations in Intelligent Systems and Applications*, 2011, pp. 107–111.

[8] H. Cao, M. Wang, X. Wang, and P. Zhu, "A Packet-Based Anomaly Detection Model for Inter-domain Routing," in *2009 IEEE International Conference on Networking, Architecture, and Storage*, 2009, pp. 192–195.

[9] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 582–592, 2000.

[10] M. Wubbeling, T. Elsner, and M. Meier, "Inter-AS routing anomalies: Improved detection and classification," in *2014 6th International Conference On Cyber Conflict (CyCon 2014)*, 2014, pp. 223–238.

[11] M. Wählisch, O. Maennel, and T. C. Schmidt, "Towards detecting BGP

route hijacking using the RPKI," in *ACM SIGCOMM Computer Communication Review*, 24-Sep-2012, vol. 42, no. 4, p. 103.

[12] H. Balakrishnan, *How YouTube was " Hijacked ,"* no. May. 2009.

[13] C. D. Marsan, "Six worst Internet routing attacks." [Online]. Available: http://www.networkworld.com/news/2009/011509-bgp-attacks.html. [Accessed: 28-Jan-2014].

[14] K. Zhang, A. Yen, X. Zhao, D. Massey, S. Felix Wu, and L. Zhang, "On Detection of Anomalous Routing Dynamics in BGP," in *Proceedings of the International IFIP-TC6 Networking Conference 2004*, vol. 5, 2004, pp. 259–270.

[15] T. Kathiravelu, A. Pears, and N. Ranasinghe, "Connectivity Models : A New Approach to Modeling Contacts in Opportunistic Networks," *Proc. Eighth Int. Inf. Technol. Conf. 2006*, p. 185, 2006.

[16] ARIN, "Regional Internet Registries," 2015. [Online]. Available: https://www.arin.net/knowledge/rirs.html. [Accessed: 08-Apr-2015].

[17] MathWorks, "Network Analysis and Visualisation," 2015. [Online]. Available: http://uk.mathworks.com/help/bioinfo/network-analysis-and-visualization.html. [Accessed: 10-Nov-2015].

[18] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, Jan. 2008.

[19] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, “Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?,” *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, 2014.

[20] G. Kaur and A. Chhabra, “Improved J48 Classification Algorithm for the Prediction of Diabetes,” *Int. J. Comput. Appl.*, vol. 98, no. 22, pp. 13–17, 2014.